

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-275066

(43)Date of publication of application : 05.10.2001

(51)Int.Cl.

H04N 5/76
H04N 5/225
H04N 5/765
H04N 5/781
H04N 7/173
// H04N101:00

(21)Application number : 2000-089304

(71)Applicant : CANON INC

(22)Date of filing : 28.03.2000

(72)Inventor : TATEYAMA JIRO

(54) IMAGE PROCESSOR, AND ITS METHOD AND STORAGE MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To solve the problem that there do not exist criteria for deciding the use of which device for which processing is more efficient for printing when each of the devices has a series of processing functions.

SOLUTION: When direct printing is carried out between an image supply device such as a digital camera 101 connected by a 1394 serial bus 103 and a printing device such as a printer 102, the devices are adaptively put in partial charge of image processing according to the relation among the processing capability by image processing functions that the devices have, data size, and data transfer rates, so that the direct printing is performed with efficiency.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-275066

(P2001-275066A)

(43) 公開日 平成13年10月5日 (2001. 10. 5)

(51) Int.Cl. ⁷	識別記号	F I	テ-マコード [*] (参考)
H 0 4 N	5/76	H 0 4 N	5/76 E 5 C 0 2 2
	5/225		5/225 F 5 C 0 5 2
	5/765		7/173 6 2 0 Z 5 C 0 6 4
	5/781	101: 00	
7/173	6 2 0	5/781	5 1 0 C
審査請求 未請求 請求項の数24 O L (全 19 頁) 最終頁に続く			

(21) 出願番号 特願2000-89304(P2000-89304)

(22) 出願日 平成12年3月28日 (2000. 3. 28)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 立山 二郎

東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康徳 (外2名)

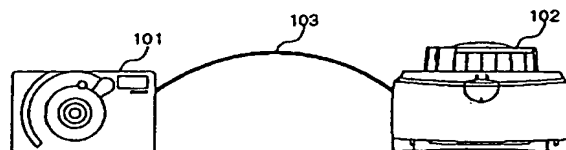
最終頁に続く

(54) 【発明の名称】 画像処理装置およびその方法、並びに、記憶媒体

(57) 【要約】

【課題】 ダイレクトプリントに対応した個々のデバイスが一連の処理機能をもつ場合に、どちらのデバイスでどの処理を行えば、効率のよいプリントを行えるかという判断基準が現状はない。

【解決手段】 1394シリアルバス103で接続されたデジタルカメラ101などの画像供給デバイスおよびプリンタ102などの印刷デバイスの間でダイレクトプリントを行う際に、それぞれのデバイスがもつ画像処理機能ごとの処理能力、並びに、データサイズおよびデータ転送速度の関係に応じて適応的に画像処理をデバイス間で分担し、効率のよいダイレクトプリントを行う。



【特許請求の範囲】

【請求項1】 シリアルバスで接続された印刷装置にデータを供給する画像処理装置であって、画像データを印刷データへ変換する画像処理手段と、前記印刷装置との間で画像処理の分担を決定する決定手段とを有することを特徴とする画像処理装置。

【請求項2】 シリアルバスで接続されたデータ供給装置からデータの供給を受ける画像処理装置であって、画像データを印刷データへ変換する画像処理手段と、前記データ供給装置との間における画像処理の分担を決定する決定手段とを有することを特徴とする画像処理装置。

【請求項3】 さらに、画像処理の能力情報を相手装置から取得する取得手段を有することを特徴とする請求項1または請求項2に記載された画像処理装置。

【請求項4】 前記能力情報は、画像処理の段階ごとに提供されることを特徴とする請求項3に記載された画像処理装置。

【請求項5】 前記決定手段は、前記能力情報、並びに、画像データサイズおよびデータ転送速度に基づき、画像処理の分担を決定することを特徴とする請求項3または請求項4に記載された画像処理装置。

【請求項6】 前記決定手段は、データ転送効率が向上するように画像処理の分担を決定することを特徴とする請求項5に記載された画像処理装置。

【請求項7】 さらに、決定された画像処理の分担を相手装置へ通知する通知手段を有することを特徴とする請求項1から請求項6の何れかに記載された画像処理装置。

【請求項8】 前記画像データを供給する装置は画像取得デバイスであり、前記データの供給を受ける装置はプリンタであることを特徴とする請求項1から請求項7の何れかに記載された画像処理装置。

【請求項9】 前記シリアルバスはIEEE1394規格に適合または準拠することを特徴とする請求項1から請求項8の何れかに記載された画像処理装置。

【請求項10】 前記能力情報は、前記IEEE1394規格で規定されたコンフィグレーションROMの情報を読み取ることによって決定されることを特徴とする請求項9に記載された画像処理装置。

【請求項11】 前記シリアルバスはUSB規格に適合または準拠することを特徴とする請求項1から請求項8の何れかに記載された画像処理装置。

【請求項12】 シリアルバスで接続された印刷装置にデータを供給する画像処理方法であって、画像データを印刷データへ変換する画像処理の各ステップを前記印刷装置との間で分担することを特徴とする画像処理方法。

【請求項13】 シリアルバスで接続されたデータ供給装置からデータの供給を受ける画像処理方法であって、画像データを印刷データへ変換する画像処理の各ステッ

ブを前記データ供給装置との間で分担することを特徴とする画像処理方法。

【請求項14】 さらに、画像処理の能力情報を相手装置から取得することを特徴とする請求項12または請求項13に記載された画像処理方法。

【請求項15】 前記能力情報は、画像処理の段階ごとに提供されることを特徴とする請求項14に記載された画像処理方法。

【請求項16】 前記分担は、前記能力情報、並びに、画像データサイズおよびデータ転送速度に基づき決定されることを特徴とする請求項14または請求項15に記載された画像処理方法。

【請求項17】 前記分担は、データ転送が一度になるように決定されることを特徴とする請求項16に記載された画像処理方法。

【請求項18】 さらに、決定された画像処理の分担を相手装置へ通知することを特徴とする請求項12から請求項17の何れかに記載された画像処理方法。

【請求項19】 前記画像データを供給する装置は画像取得デバイスであり、前記データの供給を受ける装置はプリンタであることを特徴とする請求項12から請求項18の何れかに記載された画像処理方法。

【請求項20】 前記シリアルバスはIEEE1394規格に適合または準拠することを特徴とする請求項12から請求項19の何れかに記載された画像処理方法。

【請求項21】 前記能力情報は、前記IEEE1394規格で規定されたコンフィグレーションROMの情報を読み取ることによって決定されることを特徴とする請求項20に記載された画像処理方法。

【請求項22】 前記シリアルバスはUSB規格に適合または準拠することを特徴とする請求項12から請求項19の何れかに記載された画像処理方法。

【請求項23】 シリアルバスで接続された印刷装置にデータを供給する画像処理のプログラムコードが記憶された記憶媒体であって、前記プログラムコードは少なくとも、

画像データを印刷データへ変換する画像処理の各ステップを前記印刷装置との間で分担するステップのコードを有することを特徴とする記憶媒体。

【請求項24】 シリアルバスで接続されたデータ供給装置からデータの供給を受ける画像処理のプログラムコードが格納された記憶媒体であって、画像データを印刷データへ変換する画像処理の各ステップを前記データ供給装置との間で分担するステップのコードを有することを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は画像処理装置およびその方法、並びに、記憶媒体に関し、例えば、シリアルバスで接続されたデバイス間における画像処理装置およ

びその方法に関する。

【0002】

【従来の技術】デジタルカメラで撮影した画像をプリントするには、RS-232Cなどのシリアルインタフェースやメモ리카ードを経由して画像データをパーソナルコンピュータに転送し、パーソナルコンピュータにより、画像データにプリンタの印刷形式に合った加工を施し、セントロニクスなどのパラレルインタフェースやUSB(Universal Serial Bus)などのシリアルインタフェースを介してプリンタに印刷データを送ってプリントする、という手順が必要である。

【0003】デジタルカメラのユーザが、既にパーソナルコンピュータを所有している場合は、必要なアプリケーションソフトを使って画像処理を行い、パーソナルコンピュータに接続されたプリンタで印刷する、という一般的なシステム構成である。しかし、デジタルカメラを購入してもパーソナルコンピュータを所有しない場合は、撮影した画像を印刷する手段が無いので、家庭用テレビのビデオ端子にデジタルカメラを繋いでテレビ画面上で撮影した画像を観るという使い方ができない。

【0004】また、そのようなデジタルカメラのユーザに対して、ビデオプリンタという形態のプリンタシステムが存在する。このプリンタシステムは、パーソナルコンピュータを介さずに、独自規格のシリアルインタフェース、赤外線インタフェースまたはメモ리카ードなどを使って、直接、デジタルカメラからプリンタへ画像データを転送し、プリンタ内部で画像処理を行ってプリントするものである。なお、パーソナルコンピュータを介させずに、画像データをプリンタへ転送してプリントすることを「ダイレクトプリント」と呼ぶ。

【0005】

【発明が解決しようとする課題】ダイレクトプリントを行うプリントシステムは、JPEG(Joint Photographic Experts Group)圧縮された画像データを印刷データへ変換する処理をデバイス(デジタルカメラまたはプリンタ)が行う必要があり、デバイスのデータ処理能力によって印刷時間が大きく左右される。

【0006】例えば、一般的な汎用プリンタにおいては、パーソナルコンピュータにより画像データをプリンタ固有の印刷データに変換する処理が行われることが前提になっているので、プリンタ側に高度なデータ処理能力をもたせず、コストを優先にした仕様になっている。

【0007】ダイレクトプリントに対応したプリンタでは、画像データを印刷データに変換するための処理能力を高めるために、プリンタに搭載されたCPUの高速化、CPUを複数個用いる分散処理、内蔵メモリの容量を増やして一度に変換できるデータサイズを大きくする、などの機能アップが必要になる。

【0008】また、ダイレクトプリントに対応した画像

供給デバイス(例えばデジタルカメラ)にも印刷データの作成機能をもたせる必要がある。しかし、汎用のデジタルカメラなどは、画像データの圧縮、伸長および表示機能などについては十分に高い能力を有するが、印刷データを作成する能力についてはあまり高い能力をもたない。

【0009】さらに、ダイレクトプリントに対応した個々のデバイスが一連の処理機能をもつ場合に、どちらのデバイスでどの処理を行えば、効率のよいプリントを行えるかという判断基準が、現状ではないという問題もある。

【0010】本発明は、上述の問題を解決するためのものであり、個々のデバイスに画像処理を分担することで効率的な画像処理を行うことを目的とする。

【0011】

【課題を解決するための手段】本発明は、前記の目的を達成する一手段として、以下の構成を備える。

【0012】本発明にかかる画像処理装置は、シリアルバスで接続された印刷装置にデータを供給する画像処理装置であって、画像データを印刷データへ変換する画像処理手段と、前記印刷装置との間で画像処理の分担を決定する決定手段とを有することを特徴とする。

【0013】また、シリアルバスで接続されたデータ供給装置からデータの供給を受ける画像処理装置であって、画像データを印刷データへ変換する画像処理手段と、前記データ供給装置との間における画像処理の分担を決定する決定手段とを有することを特徴とする。

【0014】本発明にかかる画像処理方法は、シリアルバスで接続された印刷装置にデータを供給する画像処理方法であって、画像データを印刷データへ変換する画像処理の各ステップを前記印刷装置との間で分担することを特徴とする。

【0015】また、シリアルバスで接続されたデータ供給装置からデータの供給を受ける画像処理方法であって、画像データを印刷データへ変換する画像処理の各ステップを前記データ供給装置との間で分担することを特徴とする。

【0016】

【発明の実施の形態】以下、本発明にかかる一実施形態のデータ転送方法を図面を参照して詳細に説明する。

【0017】図1は本発明を適用するシステムの一般的な構成例を示す図で、デジタルカメラ101およびプリンタ102をIEEE1394-1995規格のシリアルバス(以下「1394シリアルバス」と呼ぶ)で結んだダイレクトプリントシステムを示している。そこで、まずIEEE1394-1995規格(以下「IEEE1394規格」と呼ぶ)の概要を説明する。

【0018】なお、IEEE1394規格についての詳細は、1996年の8月30日にIEEE(The Institute of Electrical and Electronics Engineers, Inc.)から出版された「IEEE

Standard for a High Performance Serial Bus」に記述されている。

【0019】また、1394シリアルバスに限らず、USBのシリアルバスでデジタルカメラ101およびプリンタ102を結んでもよい。

【0020】〔概要〕図2はIEEE1394規格に準拠するデジタルインタフェース（以下「1394インタフェース」と呼ぶ）を備えるノードにより構成される通信システム（以下「1394ネットワーク」と呼ぶ）の構成例を示す図である。1394ネットワークは、シリアルデータ通信が可能

なバス型ネットワークを構成する。
【0021】図2において、ノードAからHは、IEEE1394規格に準拠した通信ケーブルを介して接続されている。これらのノードAからHは、例えば、PC(Personal Computer)、デジタルVTR(Video Tape Recorder)、DVD(Digital Video Disc)プレーヤ、デジタルカメラ、ハードディスクおよびモニタなどの電子機器である。

【0022】1394ネットワークの接続方式は、ディジーチェーン方式およびノード分岐方式に対応し、自由度の高い接続が可能である。

【0023】また、1394ネットワークでは、例えば、既存の機器がネットワークから分離されたり、新たな機器がネットワークに追加されたり、既存の機器の電源がオンオフされたりした場合、自動的にバスリセットが行われる。このバスリセットによって、1394ネットワークでは、自動的に、新たなネットワーク接続構成を認識するとともに、各機器へID情報を割り当てることができる。言い換えれば、この機能によって、1394ネットワークは、ネットワーク接続構成を常時認識することができる。

【0024】また、1394ネットワークは、他の機器から転送されたデータを中継する機能を有し、この機能によって、すべての機器が1394シリアルバスの動作状況を把握することができる。

【0025】また、1394ネットワークは、Plug&Playと呼ばれる機能を有し、この機能によって、ネットワーク接続されたすべての機器の電源をオフすることなく、ネットワークに機器を接続するだけで、自動的に接続された機器を認識する。

【0026】また、1394ネットワークは、100/200/400Mbpsのデータ転送速度に対応する。上位のデータ転送速度をもつ機器は、下位のデータ転送速度をサポートすることができるため、異なるデータ転送速度に対応する機器同士を接続することができる。

【0027】さらに、1394ネットワークは、二つの異なるデータ転送方式（すなわち、アシンクロナス(Asynchronous, 非同期)転送モードおよびアイソクロナス(Isynchronous, 同期)転送モードに対応している。アシンクロナス転送モードは、必要に応じて非同期転送することが要求されるデータ、すなわちコントロール信号やデ

ータファイルなどを転送する際に有効である。また、アイソクロナス転送モードは、所定量のデータを一定のデータレートで連続的に転送することが要求されるデータ、すなわちビデオデータやオーディオデータなどを転送する際に有効である。

【0028】アシンクロナス転送モードおよびアイソクロナス転送モードは、各通信サイクル（通常、1サイクルは125 μ s）内において混在させることが可能である。各転送モードは、サイクルの開始を示すサイクルスタートパケット(CSP)の転送後に実行される。なお、各通信サイクル期間において、アイソクロナス転送モードは、アシンクロナス転送モードよりも優先順位が高く設定されている。また、アイソクロナス転送モードの転送帯域は、各通信サイクル内で保証されている。

【0029】〔アーキテクチャ〕図3は1394インタフェースの構成要素を説明する図である。

【0030】1394インタフェースは、機能的に複数のレイヤから構成されている。1394インタフェースは、IEEE1394規格に準拠する通信ケーブル301を介して、他のノードの1394インタフェースと接続される。また、1394インタフェースは、一つ以上の通信ポート302を有し、通信ポート302はハードウェアに含まれるフィジカルレイヤ303に接続されている。

【0031】ハードウェアは、フィジカルレイヤ303およびリンクレイヤ304から構成されている。フィジカルレイヤ303は、他のノードとの物理的、電気的なインタフェース、バスリセットの検出とそれに伴う処理、入出力信号の符号化/復号、バス使用権の調停などを行う。また、リンクレイヤ304は、通信パケットの生成とその送受信、サイクルタイマの制御などを行う。

【0032】また、図3において、ファームウェアは、トランザクションレイヤ305およびシリアルバスマネージメント306を含んでいる。トランザクションレイヤ305は、アシンクロナス転送モードを管理し、各種のトランザクション（リード、ライトおよびロック）を提供する。シリアルバスマネージメント306は、後述するCSRアーキテクチャに基づいて、自ノードの制御、自ノードの接続状態およびID情報の管理、並びに、1394ネットワークの資源管理を行う機能を提供する。

【0033】以上のハードウェアおよびファームウェアが実質的に1394インタフェースを構成し、それらの基本構成はIEEE1394規格により規定されている。

【0034】また、ソフトウェアに含まれるアプリケーションレイヤ307は、使用するアプリケーションソフトウェアによって異なり、1394ネットワーク上でどのようにデータを通信するかを制御する。例えば、デジタルVTRの動画像データの場合はAV/Cプロトコルなどの通信プロトコルが規定されている。

【0035】●リンクレイヤ

図4はリンクレイヤ304が提供可能なサービスを示す図で

ある。

【0036】リンクレイヤ304は次の四つのサービスを提供する。なお、リンク応答(LK_DATA.response)は、ブロードキャスト通信およびアイソクロナスパケットを転送する場合には存在しない。

(1)リンク要求(LK_DATA.request) 応答ノードに所定パケットの転送を要求する

(2)リンク通知(LK_DATA.indication) 応答ノードへ所定パケットの受信を通知する

(3)リンク応答(LK_DATA.response) 応答ノードからアク
ノリッジを送信する

(4)リンク確認(LK_DATA.confirmation) 要求ノードへアク
ノリッジ受信を通知する

【0037】また、リンクレイヤ304は、上述のサービスに基づいて、上述の二種類の転送方式、すなわちアシンクロナス転送モードおよびアイソクロナス転送モードを実現する。

【0038】●トランザクションレイヤ

図5はトランザクションレイヤ305が提供可能なサービスを示す図である。

【0039】トランザクションレイヤ305は次の四つのサービスを提供する。

(1)トランザクション要求(TR_DATA.request) 応答ノードに所定のトランザクションを要求する

(2)トランザクション通知(TR_DATA.indication) 応答ノードへ所定のトランザクション要求の受信を通知する

(3)トランザクション応答(TR_DATA.response) 応答ノードからのトランザクション状態情報(ライト/ロックの場合はデータを含む)

(4)トランザクション確認(TR_DATA.confirmation) 要求ノードへトランザクション状態情報の受信を通知する

【0040】また、トランザクションレイヤ305は、上述のサービスに基づいて、アシンクロナス転送を管理し、次の三種類のトランザクション、すなわちリードトランザクション、ライトトランザクション、および、ロックトランザクションを実現する。

(1)リードトランザクション: 要求ノードが応答ノードの特定アドレスに格納された情報を読み取る

(2)ライトトランザクション: 要求ノードが応答ノードの特定アドレスに所定の情報を書き込む

(3)ロックトランザクション: 要求ノードから応答ノードへ、参照データおよび更新データを転送し、応答ノードの特定アドレスの情報および参照データを比較して、その比較結果に応じて特定アドレスの情報を更新データに書き換える

【0041】●シリアルバスマネージメント

シリアルバスマネージメント306は次の三つの機能を提供する。

(1)ノード制御: 上述の各レイヤを管理し、他のノードとの間で実行されるアシンクロナス転送を管理する機能

を提供する

(2)アイソクロナスリソースマネージャ(IRM): 他のノードとの間で実行されるアイソクロナス転送を管理する機能を提供する。

(3)バスマネージャ: IRMの機能を有し、IRMよりも高度なバス管理機能を提供する

【0042】IRMは、具体的には、転送帯域幅およびチャネル番号の割り当てに必要な情報を管理し、これらの情報を他のノードに対して提供する。IRMは、ローカルバス上に唯一存在し、バスリセットの度に他の候補者(IRMの機能を有するノード)の中から動的に選出される。また、IRMは、バスマネージャが提供可能な機能(接続構成の管理、電源管理および速度情報の管理など)の一部を提供してもよい。

【0043】バスマネージャは、具体的には、より高度な管理、それら管理情報に基づく1394シリアルバスの最適化などを行い、さらに、それらの情報を他のノードに提供する機能を有する。より高度な管理とは、通信ケーブルを介して電力を供給することが可能か否か、電源の供給が必要か否かなどの情報をノードごとに管理すること(高度な電源管理)、各ノード間の最大転送速度を管理すること(高度な速度情報の管理)、並びに、トポロジマップを作成すること(高度な接続構成の管理)である。

【0044】また、バスマネージャは、1394ネットワークを制御するためのサービスをアプリケーションに対して提供することができる。このサービスには次の三つなどがある。

(1)シリアルバス制御要求(SB_CONTROL.request) アプリケーションがバスリセットを要求するサービス

(2)シリアルバスイベント制御確認(SB_CONTROL.confirmation) アプリケーションに対してシリアルバス制御要求を確認するサービス

(3)シリアルバスイベント通知(SB_CONTROL.indication) 非同期に発生するイベントをアプリケーションに対して通知するサービス

【0045】【アドレス指定】図6は1394インタフェースにおけるアドレス空間を説明する図である。なお、1394インタフェースは、ISO/IEC 13213:1994に準じたCSR(Command and Status Register)アーキテクチャに従い、64ビット幅のアドレス空間を規定している。

【0046】図6において、最初の10ビットのフィールド601は所定の1394シリアルバスを指定する番号に使用され、次の6ビットのフィールド602は所定の機器(ノード)を指定する番号に使用される。これら上位16ビットを「ノードID」と呼び、各ノードはこのノードIDにより他のノードを識別する。また、各ノードは、このノードIDを用いて相手を識別した通信を行うことができる。

【0047】残りの48ビットからなるフィールドは、各ノードが備えるアドレス空間(256Mバイト構造)に対応

し、その内の20ビットのフィールド603はアドレス空間を構成する複数の領域を指定する。フィールド603の「0から0xFFFFD」の領域はメモリ空間、「0xFFFFE」の領域はプライベート空間と呼ばれ各ノードが自由に利用することができるアドレス空間である。また、「0xFFFF」の領域はレジスタ空間と呼ばれ、バスに接続されたノード間において共通の情報が格納される。各ノードは、レジスタ空間に格納された情報を用いることで、ノード間の通信を管理することができる。

【0048】最後の28ビットのフィールド604は、各ノードにおいて共通あるいは固有の情報が格納されるアドレスを指定する。例えば、レジスタ空間において、最初の512バイトは、CSRアーキテクチャのコア（CSRコア）レジスタとして使用される。CSRコアレジスタに格納される情報のアドレスおよび機能を図7に示す。図7に示すオフセット値は「0xFFFF00000000」からの相対位置である。

【0049】続く512バイトは、シリアルバス用のレジスタとして使用される。シリアルバスレジスタに格納される情報のアドレスおよび機能を図8に示す。図8に示すオフセット値は「0xFFFF0000200」からの相対位置である。

【0050】続く1024バイトはコンフィグレーション(Configuration)ROMに使用される。「0xFFFF0000400」から配置されるコンフィグレーションROMには最小形式および一般形式がある。最小形式のコンフィグレーションROMの構成を図9に示す。図9において、ベンダIDは、IEEEにより、各ベンダに対して固有に割り当てられた24ビットの数値である。また、一般形式のコンフィグレーションROMの構成を図10に示す。図10において、上述のベンダIDは、Root Directory1002に格納されている。Bus Info Block1001およびRoot Leaf1005には、各ノードを識別する固有のID情報としてノードユニークIDを保持することが可能である。

【0051】ノードユニークIDは、メーカーや機種によらず、一つのデバイスを特定することができる固有のIDを定めるように規定されている。ノードユニークIDは64ビットで構成され、上位24ビットは上述のベンダIDを示し、下位48ビットは各デバイスの製造者が自由に設定可能な情報（例えばデバイスの製造番号など）を示す。なお、このノードユニークIDは、例えば、バスリセットの前後で、継続して特定のデバイスを認識する場合に利用される。

【0052】また、図10に示すRoot Directory1002にはデバイスの基本的な機能に関する情報を保持させることが可能である。詳細な機能情報は、Root Directory1002からオフセットされるサブディレクトリUnit Directories1004に格納される。Unit Directories1004には、例えば、デバイスがサポートするソフトウェアユニットに関する情報が格納される。具体的には、ノード間のデータ

通信を行うためのデータ転送プロトコル、所定の通信手順を定義するコマンドセットなどに関する情報が保持される。

【0053】また、図10に示すNode Dependent Info Directory1003にはデバイス固有の情報を保持させることが可能である。Node Dependent Info Directory1003は、Root Directory1002によりオフセットされる。

【0054】さらに、図10に示すVendor Dependent Information1006にはノードを製造あるいは販売するベンダ固有の情報を保持させることができる。

【0055】図6に示すフィールド604の残りの領域はユニット空間と呼ばれ、各デバイスに固有の情報、例えば、各機器の識別情報（会社名や機種名など）や使用条件などが格納されたアドレスを指定する。ユニット空間のシリアルバス装置レジスタに格納される情報のアドレスおよび機能を図11に示す。図11に示すオフセット値は「0xFFFF0000800」からの相対位置である。

【0056】なお、一般に、異種のバスシステムの設計を簡略化したい場合、各デバイスは、ユニット空間の最初の2048バイトのみを使うべきである。つまり、CSRコアレジスタ、シリアルバスレジスタおよびコンフィグレーションROMの2048バイトと、ユニット空間の最初の2048バイトとを合わせた4096バイトで構成することが望ましい。

【0057】[通信ケーブル] 図12はIEEE1394規格に準拠した通信ケーブルの断面図である。

【0058】通信ケーブルは、二組のツイストペア信号線および電源ラインで構成されている。通信ケーブルに電源ラインを設けることにより、1394インタフェースは、主電源がオフの機器、故障によって電力が低下した機器などにも電力を供給することができる。なお、電源ラインによって供給可能な直流電力は8から40V、最大1.5Aと規定されている。

【0059】二組のツイストペア信号線は、DS-Link(Data/Strobe Link)符号化方式によって符号化された情報信号を伝送する。図13はDS-Link符号化方式を説明する図である。

【0060】DS-Link符号化方式は、高速なシリアルデータ通信に適し、その構成は二組のツイストペア信号線を必要とする。一組の信号線はデータ信号を送り、もう一組の信号線はストロブ信号を送る。受信側は、二組の信号線によって受信されるデータ信号およびストロブ信号の排他的論理和をとることによって、クロックを生成することができる。DS-Link符号化方式を用いる1394インタフェースは、例えば次のような利点を有する。

- (1)他の符号化方式に比べて転送効率が高い
- (2)PLL(Phase Locked Loop)回路が不要になり、コントローラLSIの回路規模を小さくできる
- (3)アイドル状態を示す信号の送信が不要になるからトランシーバ回路をスリープ状態にし易く、消費電力の低

減が図れる

【0061】[バスリセット]各ノード(正しくはノードの1394インタフェース)は、ネットワークの接続構成に変化が生じたことを自動的に検出することができる。この場合、1394ネットワークは、以下に示す手順により、バスリセットと呼ばれる処理を行う。なお、接続構成の変化は、各ノードが備える通信ポートに加わるバイアス電圧の変化により検知される。

【0062】ネットワークの接続構成の変化、例えばノードの挿抜、ノードの電源のオンオフなどによるノード数の増減などを検出したノード、または、新たな接続構成を認識する必要のあるノードは、1394インタフェースを介して、1394シリアルバスにバスリセット信号を送信する。

【0063】バスリセット信号を受信したノードは、バスリセットの発生を自身のリンクレイヤ304に伝達するとともに、そのバスリセット信号を他のノードに転送する。バスリセット信号を受信したノードは、今まで認識していたネットワークの接続構成および各機器に割り当てられたノードIDをクリアにする。最終的にすべてのノードがバスリセット信号を検知した後、各ノードは、バスリセットに伴う初期化処理、すなわち、新たな接続構成の認識および新たなノードIDの割り当てを自動的に行う。

【0064】なお、バスリセットは、先に述べたような接続構成の変化により起動されるほかに、ホスト側の制御によって、アプリケーションレイヤ307がフィジカルレイヤ303に直接命令を出すことによって起動される場合もある。また、バスリセットが起動されるとデータ転送は一時中断され、バスリセットに伴う初期化処理が終了した後、新しいネットワークの接続構成の下で再開される。

【0065】●バスリセットのシーケンス

上述したように、バスリセットが起動された後、各ノードは、新たな接続構成の認識および新たなノードIDの割り当てを自動的に実行する。以下、バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを図14から図16を用いて説明する。

【0066】図14は、図2に示す1394ネットワークにおけるバスリセット起動後の状態を説明する図である。図14において各ノードの通信ポートは、ノードA EおよびFは一つ、ノードBおよびCは二つ、ノードDは三つである。各通信ポートには、各ポートを識別するためのポート番号が付されている。

【0067】以下、図14に示すネットワークの接続構成においてバスリセットが開始され、ノードIDが割り当てられるまでを、図15に示すフローチャートを用いて説明する。

【0068】ステップS1501において、1394ネットワークを構成する各ノードAからFは、バスリセットの発生を

常時監視し、接続構成の変化を検出したノードからバスリセット信号が出力されると、各ノードは以下の処理を実行する。

【0069】バスリセットが発生すると、ステップS1502で、各ノードはそれぞれが備える通信ポート間において親子関係の宣言を行う。親子関係の宣言は、ステップS1503で、すべてのポート間の親子関係が決定されたと判定されるまで繰り返される。

【0070】すべてのポート間の親子関係が決定すると、ステップS1504で、ネットワークの調停を行うノード、すなわちルートが決定される。次に、ステップS1505で、各ノードの1394インタフェースは、自身のノードIDを自動的に設定する作業を実行する。ノードIDの設定は、ステップS1506で、すべてのノードのノードIDが設定されたと判定されるまで繰り返される。

【0071】すべてのノードのノードIDが設定されると、ステップS1507で、各ノードはアイソクロナス転送あるいはアシンクロナス転送によるデータ転送を実行する。図15には、ステップS1507の処理が終了した後にステップS1501の処理が実行されるように記載したが、正しくは、各ノードはステップS1507のデータ転送を実行するとともに、ステップS1501でバスリセットの発生を監視することになる。そして、バスリセットが発生すると、各ノードは、データ転送を停止して、ステップS1502からS1506の処理を実行した後、新たな接続構成の下でデータ転送を再開することになる。

【0072】以上の手順により、各ノードの1394インタフェースは、バスリセットが発生する度に、新たな接続構成の認識、および、新たなノードIDの割り当てを自動的に実行することができる。

【0073】●親子関係の決定

図16はステップS1502の親子関係の宣言、つまり各ポート間の親子関係を認識する処理を詳細に示すフローチャートである。

【0074】各ノードは、ステップS1601で、自分が備える通信ポートの接続状態、つまり接続か未接続かを確認し、他のノードへ接続されている通信ポート(以下「接続ポート」と呼ぶ)の数をカウントする。次に、ステップS1602で接続ポート数を判定して、接続ポート数が「1」のノードは、ステップS1603で自分が「リーフ」とであると認識する。リーフは、他の一つのノードとだけ接続されているノードのことで、図14においてはノードA EおよびFがリーフである。リーフは、ステップS1604で、その接続ポートに接続されたノードに対して「自分は子(Child)」であることを宣言する。この際、リーフは、その接続ポートが親ノードに接続された通信ポートである「親(parent)ポート」と認識する。

【0075】親子関係の宣言は、まず、ネットワークの末端であるリーフと、接続ポート数が「2」以上のノードである「ブランチ」との間で行われ、続いて、ブランチ

チとブランチとの間で行われる。各通信ポート間の親子関係は、早く宣言を行った通信ポートから順に決定される。また、子であることを宣言した通信ポートは「親ポート」とであると認識され、その宣言を受け付けた通信ポートは子ノードに接続された通信ポートである「子ポート」と認識される。例えば、図14において、ノードA、EおよびFは、自分がリーフであると認識した後、親子関係の宣言を行い、ノードA-D間、ノードE-D間およびノードF-D間は「子-親」と決定される。

【0076】一方、接続ポートの数が「2」以上のノードは、ステップS1605で自分をブランチであると認識する。ブランチは、ステップS1606で、その接続ポートに接続されたノードから親子関係の宣言を受け付ける。宣言を受け付けた接続ポートは、上述したように「子ポート」として認識される。一つの接続ポートを「子ポート」と認識した後、ブランチは、ステップS1607で、未だ親子関係が決定されていない接続ポート（以下「未定義ポート」と呼ぶ）の数を調べて、未定義ポート数が二つ以上の場合はステップS1606の親子関係宣言の受け付けを繰り返す。

【0077】ブランチは、未定義ポート数が「1」以下になると、ステップS1608の判定により未定義ポート数が「1」の場合は、ステップS1609で、その通信ポートを「親ポート」と認識して、その通信ポートに接続されたノードに対して「自分は子」を宣言する。ブランチは、未定義ポート数が「1」になるまで親子関係を宣言することができない。例えば、図14において、ノードB、CおよびDは、自分がブランチであると認識するとともに、リーフあるいは他のブランチから親子関係の宣言を受け付ける。ノードDは、D-E間およびD-F間の親子関係が決定された後、ノードCに対して親子関係の宣言を行うことができる。そして、ノードDから親子関係の宣言を受けたノードCは、ノードBに対して親子関係の宣言を行うことができる。

【0078】また、ステップS1608の判定時に未定義ポートが存在しない場合、つまりブランチが備えるすべての接続ポートが「子ポート」になった場合、そのブランチは、ステップS1610で、自分が「ルート」とであると認識する。例えば、図14において、接続ポートのすべてが親ポートとなったノードBは、1394ネットワーク上の通信を調停するルートとして他のノードに認識される。図14にはノードBがルートに決定される例を示したが、ノードBが親子関係を宣言するタイミングによっては、他のブランチまたはリーフがルートになる可能性がある。すなわち、接続構成および親子関係を宣言するタイミングによって、どのノードもルートになる可能性があり、たとえ同じ接続構成であったとしても、いつも同じノードがルートになるとは限らない。

【0079】このようにして、すべての接続ポートの親子関係が宣言されると、各ノードは、ステップS1611

で、1394ネットワークの接続構成を階層構造（ツリー構造）として認識することができる。なお、親ノードは階層構造における上位であり、子ノードは階層構造における下位になる。

【0080】●ノードIDの割当

図17Aおよび17BはステップS1505のノードIDの設定、つまり各ノードにノードIDを割り当てる処理を詳細に示すフローチャートで、図17Aはルートの処理、図17Bはルート以外の処理を示している。ノードIDは、上述したようにバス番号およびノード番号から構成されるが、本実施形態では、各ノードが同一バス上に存在するものとして、各ノードには同一のバス番号が割り当てられるものとする。

【0081】ルートは、ステップS1701で、ノードIDが未設定のノードが接続されている子ポートの内、ポート番号が最小の通信ポートに接続されたノードに対してノードIDの設定許可を与える。次に、ルートは、ステップS1702で、子ポートに接続された全ノードのノードIDが設定されたか否かを判定し、未設定のノードがあればステップS1701を繰り返す。つまり、ルートは、最小のポート番号をもつ通信ポートに接続された全ノードのノードIDが設定された後、その子ポートを設定済とし、続いて、次に小さいポート番号をもつ通信ポートに接続されたノードに対して同様の制御を行う。

【0082】最終的に、子ポートに接続された全ノードのノードIDが設定されると、ルートは、ステップS1703で自分のノードIDを設定し、ステップS1704で、後述するセルフIDパケットをブロードキャストする。なお、ノードIDに含まれるノード番号は、基本的にリーフ、ブランチの順に0、1、2、…と割り当てられる。従って、ルートが最も大きなノード番号を有することになる。

【0083】一方、ノードIDの設定許可をルートから得たノードは、ステップS1711で、ノードIDが未設定のノードを含む子ポートがあるか否かを判定し、そのような子ポートがある場合はステップS1712で、その子ポートに接続されたノードに対してノードIDの設定許可を与える。ここでノードIDの設定許可を得たノードも図17Bの処理を実行することになる。

【0084】そして、ノードは、ステップS1713で、再び、ノードIDが未設定のノードを含む子ポートがあるか否かを判定する。ステップS1711またはS1713でノードIDが未設定のノードを含む子ポートはないと判定されると、ノードは、ステップS1714で、自分のノードIDを設定し、ステップS1715で、自分のノード番号および通信ポートの接続状態に関する情報などを含むセルフIDパケットをブロードキャストする。

【0085】ブロードキャストとは、あるノードの通信パケットを1394ネットワークを構成する不特定多数の他のノードすべてに転送することである。各ノードは、セルフIDパケットを受信することによって、各ノードに割

り当てられたノード番号を認識することができ、自分に割り当て可能なノード番号を知ることができる。

【0086】例えば、図14において、ルートであるノードBは、まずポート番号が最小の#0の通信ポートに接続されたノードAに対してノードIDの設定許可を与える。ノードAは、自分のノード番号として「0」を割り当て、自分のノードIDを設定した後、そのノードIDを含むセルフIDパケットをブロードキャストする。

【0087】次に、ルートは、ポート番号#1の通信ポートに接続されたノードCに対してノードIDの設定許可を与える。ノードCはポート番号#2の通信ポートに接続されたノードDに対してノードIDの設定許可を与える、ノードDはポート番号#0の通信ポートに接続されたノードEに対してノードIDの設定許可を与える。ノードEのノードIDが設定されると、ノードDは、ポート番号#1の通信ポートに接続されたノードFに対してノードIDの設定許可を与える。以下説明を省略するが、このような手順で全ノードのノードIDが設定される。

【0088】●セルフIDパケット

図18はセルフIDパケットの構成例を示す図である。

【0089】1801はセルフIDパケットを送出したノードのノード番号が格納されるフィールド、1802は対応可能な転送速度に関する情報が格納されるフィールド、1803はバス管理機能の有無（バスマネージャの能力の有無など）を示すフィールド、1804は電力の消費および供給の特性に関する情報が格納されるフィールドである。また、1805から1807はそれぞれポート番号#0から#2の通信ポートの接続状態に関する情報（接続、未接続および通信ポートの親子関係など）が格納されるフィールドである。

【0090】なお、セルフIDパケットを送出するノードにバスマネージャになり得る能力がある場合はフィールド1803に示すコンテンツビットを「1」にし、その能力がなければコンテンツビットを「0」にする。

【0091】●バスマネージャ

バスマネージャとは、上述のセルフIDパケットに含まれる各種の情報に基づいて、次の管理などを行うノードである。それらの機能により、バスマネージャになるノードは1394ネットワーク全体のバス管理を行うことができる。

(1)バスの電源管理：通信ケーブルを介して電力の供給が可能か否か、電力の供給が必要か否かなどの情報をノードごとに管理する

(2)速度情報の管理：各ノードが対応可能な転送速度に関する情報から各ノード間の最大転送速度を管理する

(3)トポロジマップ情報の管理：通信ポートの親子関係情報からネットワークの接続構成を管理する

(4)トポロジマップ情報に基づくバスの最適化

(5)上記の情報を他のノードに提供する

【0092】ノードIDの設定終了後、複数のノードがバ

スマネージャの能力を備える場合、最大のノード番号をもつノードがバスマネージャになる。従って、最大のノード番号をもつルートがバスマネージャの能力を有する場合、ルートがバスマネージャになる。しかし、ルートがバスマネージャの能力を備えていない場合は、ルートの次に大きいノード番号をもち、バスマネージャの能力を備えるノードがバスマネージャになる。

【0093】また、どのノードがバスマネージャになったかについては、各ノードがブロードキャストするセルフIDパケットのコンテンツビット1803をチェックすることにより把握することができる。

【0094】〔アービトレーション〕図19は、図2に示したネットワーク構成におけるアービトレーションを説明する図である。

【0095】1394ネットワークでは、データ転送に先立ち、必ずバス使用権のアービトレーション（調停）が行われる。1394ネットワークは、論理的なバス型ネットワークであり、各ノードから転送されたパケットを他のノードに中継することによって、ネットワーク内のすべてのノードに同じパケットを転送することのできる。従って、パケットの衝突を防ぐために必ずアービトレーションが必要になる。これによって、あるタイミングにおいては一つのノードがデータ転送を行うことができる。

【0096】図19(a)はノードBおよびFがバスの使用権を要求している状態を示す図である。アービトレーションが始まるとノードBおよびFは、それぞれ親ノードに向かって、バス使用権を要求する。ノードBの要求を受けた親ノードであるノードCは、自分の親ノードでありルートであるノードDへバス使用権の要求を中継する。つまり、バス使用権の要求は、最終的に、アービトレーションを行うルートに届けられる。

【0097】バス使用権の要求を受けたルートは、どのノードにバス使用権を与えるかを決める。アービトレーションはルートのみが行え、アービトレーションに勝ったノードにはバス使用権が与えられる。

【0098】図19(b)はノードFにバス使用権が与えられ、ノードBの要求が拒否された状態を示す図である。アービトレーションに負けたノードに対してルートは、DP(Data Prefix)パケットを送り、要求が拒否されたことを知らせる。要求を拒否されたノードは、次のアービトレーションで再びバス使用権を要求し、バス使用権が与えられるまでバスの使用（データ転送）を待機する。

【0099】このようにして、アービトレーションを行うことにより、ルートは、1394ネットワークのバスの使用を管理する。

【0100】〔通信サイクル〕各通信サイクルの期間内において、アイソクロナス転送モードおよびアシンクロナス転送モードを時分割に混在させることができる。通信サイクルの一期間は、通常125μsである。図20は一通

10

20

40

50

信サイクルにアイソクロナス転送モードおよびアシンクロナス転送モードを混在させた状態を説明する図である。

【0101】アイソクロナス転送は、アシンクロナス転送に優先して実行される。その理由は、サイクルスタートパケット(CSP)の後、アシンクロナス転送を起動するために必要なアイドル期間(subaction gap)が、アイソクロナス転送を起動するため必要なアイドル期間(isochronous gap)よりも長くなるように設定されているためである。これにより、アイソクロナス転送は、アシンクロナス転送に優先して実行される。

【0102】各通信サイクルの開始時に、サイクルスタートパケット(CSP)が所定のノードから転送される。各ノードは、このCSPによりタイミング調整を行うことで、他のノードと同じ時間を計時することができる。

【0103】【アイソクロナス転送モード】アイソクロナス転送モードでは同期型のデータ転送が行われる。アイソクロナス転送は、通信サイクルの開始後、所定の期間に実行可能である。また、アイソクロナス転送モードにおいては、リアルタイム転送を維持するために、各サイクルで必ずアイソクロナス転送が実行される。

【0104】また、アイソクロナス転送モードは、とくに動画データや音声を含むサウンドデータなどのリアルタイム転送を必要とするデータの転送に適した転送モードである。アイソクロナス転送モードは、アシンクロナス転送モードのように一対一の通信ではなく、ブロードキャスト通信である。つまり、あるノードから送出されアイソクロナス転送されるパケットは、ネットワーク上のすべてのノードに一樣に転送される。なお、アイソクロナス転送にはack(受信確認用返信コード)は存在しない。

【0105】図20において、チャネルe、sおよびklは、各ノードがアイソクロナス転送を行う期間を示している。1394インタフェースは、複数の異なるアイソクロナス転送を区別するために、それぞれ異なるチャネル番号を与える。これにより、複数ノードによるアイソクロナス転送が可能になる。ただし、このチャネル番号は送信先を特定するものではなく、データに対する論理的な番号を与えているに過ぎない。

【0106】また、図20に示すアイソクロナスgapはバスのアイドル状態を示すものである。このアイドル状態が所定時間を経過した後、アイソクロナス転送を希望するノードは、バスが使用できると判断してバス使用権を要求する。

【0107】図21にアイソクロナス転送されるパケットのフォーマットを示す図である。なお、以下では、アイソクロナス転送されるパケットを「アイソクロナスパケット」と呼ぶ。アイソクロナスパケットはヘッダ部2101、ヘッダCRC2102、データ部2103およびデータCRC2104から構成される。

【0108】ヘッダ部2101には、データ部2103のデータ長(data_length)が格納されるフィールド2105、アイソクロナスパケットのフォーマット情報(tag)が格納されるフィールド2106、アイソクロナスパケットのチャネル番号(channel)が格納されるフィールド2107、パケットのフォーマットおよび実行しなければならない処理を識別するトランザクションコード(tcode)が格納されるフィールド2108、並びに、同期化コード(sv)が格納されるフィールド2109がある。

【0109】【アシンクロナス転送モード】アシンクロナス転送モードでは非同期型のデータ転送が行われる。アシンクロナス転送は、アイソクロナス転送期間の終了後、次の通信サイクルが開始されるまでの間、すなわち次のCSPが転送されるまでの間に実行可能である。

【0110】図20において、最初のサブアクションギャップ(subaction gap)は、バスのアイドル状態を示す。このアイドル時間が所定値になった後、アシンクロナス転送を希望するノードは、バスが使用できると判断して、バス使用権を要求する。アービトレーションによりバス使用権を得たノードは、アシンクロナス転送するパケットを所定のノードへ向けて送信する。このパケットを受信したノードはack(受信確認用返信コード)あるいは応答パケットをack gap後に返送する。

【0111】図22はアシンクロナス転送されるパケットのフォーマットを示す図である。なお、以下では、アシンクロナス転送されるパケットを「アシンクロナスパケット」と呼ぶ。アシンクロナスパケットは、ヘッダ部2201、ヘッダCRC2202、データ部2203およびデータCRC2204から構成される。

【0112】ヘッダ部2201には、宛先ノードのノードID(destination_ID)が格納されるフィールド2205、発信元(ソース)ノードのノードID(source_ID)が格納されるフィールド2206、一連のトランザクションを示すラベル(tl)が格納されるフィールド2207、再送ステータスを示すコード(rt)が格納されるフィールド2208、パケットのフォーマットおよび実行しなければならない処理を識別するトランザクションコード(tcode)が格納されるフィールド2209、優先順位(pri)が格納されるフィールド2210、宛先のメモリアドレス(destination_offset)が格納されるフィールド2211、データ部のデータ長(data_length)が格納されるフィールド2212、並びに、拡張されたトランザクションコード(extended_tcode)が格納されるフィールド2213がある。

【0113】また、アシンクロナス転送は、ソースノードから宛先ノードへの一対一の通信である。ソースノードから送信されたパケットは、ネットワーク中の各ノードに行き渡るが、各ノードは宛先が自分のアドレス以外を示すパケットを無視する。従って、宛先ノードのみがそのパケットを読み込むことができる。

【0114】なお、アシンクロナス転送中に次のCSPを

転送すべき時間に至った場合、無理に転送を中断せず、その転送が終了した後、次のCSPが送信される。これにより、一つの通信サイクルが125 μ s以上続いたときは、その分、次の通信サイクルの期間を短縮する。このようにすることで、1394ネットワークはほぼ一定の通信サイクルを維持することができる。

【0115】[プリンタ] 図23は、図1に示すプリンタ102の内部構成例を示すブロック図で、インクジェット方式のプリントヘッド2307をもつプリンタ装置である。

【0116】CPU2301は、ROM2303に格納された実行プログラムに従い、プリンタ102の制御を行う。RAM2302はプリンタ102の内部メモリで、インタフェースを経由してプリンタ102へ入力される画像データや印刷データを一時保存する受信領域、印刷データから変換された、プリントヘッド2507に対応したCMYK各色のインクを吐出させるためのデータを保存するデータ領域、並びに、CPU2301がデータ処理のために使うワーク領域などが存在する。また、プリンタ102内の各ブロックは、プリンタ102内のシステムバスを介して、様々なデータ転送、制御および処理を行う。

【0117】プリンタ102の基本的な動作について説明する。CPU2301は、プリンタコントローラ2304およびプリンタドライバ2305を介してモータ2306を駆動して、プリントヘッド2307が搭載されたキャリア部や紙送り機構を制御する。それとともに、インクを吐出させるためのデータをRAM2302から読み出してプリンタコントローラ2304へ送り、プリンタドライバ2305を経由して、プリントヘッド2307を駆動することで、印刷が実行される。

【0118】また、プリンタ102は、外部インタフェースとして1394インタフェースを構成するLINKチップ2308およびPHYチップ2309を備える。従って、1394ネットワークを経由して、外部デバイスから入力される画像データや印刷データに基づく可視像を記録紙に印刷することができる。

【0119】[デジタルカメラ] 図24は、図1に示すデジタルカメラ101の内部構成例を示すブロック図であるが、撮影に関する構成は除き、画像データの変換処理やデータ転送に関連する構成だけを示している。

【0120】CPU2401は、ROM2403に格納されたプログラムに従い、デジタルカメラ101の各ブロックを制御する。CPU2401は、RAM2402をワークエリアとして様々な制御や画像処理を行うとともに、RAM2402を一時保存メモリに利用してデータ転送を行う。

【0121】カメラコントローラ2406は、撮影に必要な機器制御を司り、CCDから画像を読み込んだり、撮影画像をLCDに表示したり、撮影時のフォーカスや露出の設定などを行う。撮影画像のデータは、例えば、JPEG圧縮されたデータとしてメモ리카ード2408に保存される。メモ리카ードは、通常、システムバスに直結してアクセスすることはできないので、ATAタイプのメモ리카

ードの読書制御を行うカードコントローラ2407を介してシステムバスに接続される。

【0122】また、デジタルカメラ101は、外部インタフェースとして1394インタフェースを備えていて、LINKチップ2404およびPHYチップ2405を経由して画像データや印刷データを外部デバイスに送信することができる。

【0123】[データ処理] 図25はデジタルカメラ101で撮影されJPEG圧縮された画像データを、プリンタ102に合わせた印刷データに変換する処理を示すブロック図である。

【0124】画像処理を行うには圧縮されたデータを伸長する必要があるので、JPEGデータをJPEG伸長部251で伸長してRGBデータに変換する。

【0125】RGBデータは、画像補正部252により画像のコントラスト、明るさ、ガンマ、カラーの彩色、色かぶりおよび輪郭などの補正処理が施され、補正後のデータであるRGB'データになる。

【0126】RGB'データは、色処理部253によりプリンタ102のが使用するインクによって決まる色空間に合わせて、シアン(Cyan)、マゼンタ(Magenta)およびイエロー(Yellow)を表すCMYデータに変換される。

【0127】CMYデータは、UCR部254によりブラック(Black)成分が抽出され、CMYにBlackを表すデータを加えたのCMYK四色のCMYKデータに変換される。なお、UCRは「UnderColor Removal」の略で「下色除去」とも呼ばれる。

【0128】さらに、多値のCMYKデータを、プリンタ102の印刷方式であるインクジェットプリント方式に合わせて、インクの吐出を表すデータに二値化または量子化する必要がある。ハーフトーニング部255は、多値のCMYKデータを、プリンタ102の解像度に合わせてCMYKデータに変換する。つまり、誤差拡散やディザなどの疑似階調処理を利用して、多値のCMYKデータを、プリンタ102に応じた解像度かつ二値、三値または四値などのCMYKデータに変換する。なお、図25にはCMYK二値データに変換される例を示す。

【0129】このCMYK二値データは、最終的に、プリンタ102によりプリントヘッド2307の構造に合わせた吐出パターンデータに変換され、プリントヘッド2307の駆動が行われる。

【0130】画像処理のパフォーマンスを割り出すには、データ処理時間を測定する必要がある。表1は、デジタルカメラ101およびプリンタ102により、規定の画像であるサンプルJPEGデータに図25に示す画像処理を施す場合の処理時間の一例を示している。使用するサンプル画像はとくに限定されないが、単調ではない色合いをもつ画像のJPEGデータを利用する方が、測定誤差が生じ難いと考えられる。

【0131】表1は100KバイトのサンプルJPEGデータ

を、あるデジタルカメラ101およびプリンタ102に処理させて、その処理時間を測定したものである。また、表1には、それぞれの処理部についてサンプル画像データの処理時間を示すが、これらの処理にかかった時間をそのままパフォーマンス値として用いれば、数値が小さい*

*ほど処理能力が高いことになり、なおかつ、処理時間の目安として使うことができる。

【0132】

【表1】

処理部	データサイズ (Kバイト)	処理時間(ms)	
		カメラ	プリンタ
JPEGデータ	100	—	—
JPEG伸長部	300	60	120
画像補正部	300	60	60
色処理部	450	60	40
UCR部	400	30	33
ハーフトニング部	800	2000	1000

少なくともデバイス自体の(表1に示すような)処理能力がROMに予め格納されているとすれば、1394ネットワークを介して、デジタルカメラの処理能力をプリンタへ、あるいは、プリンタの処理能力をデジタルカメラへ提供すれば、それらの情報から印刷に関する一連の画像処理をデジタルカメラおよびプリンタでどのように分担すべきかを、デジタルカメラやプリンタのCPU ※

※が判定することが可能になる。

【0133】表2は、表1に示したパフォーマンス値を用いて、10MバイトのJPEGデータを処理して印刷するのに必要な処理時間を推定した結果を示す。

【0134】

【表2】

処理部	データサイズ (Mバイト)	処理時間(s)	
		カメラ	プリンタ
JPEGデータ	10	—	—
JPEG伸長部	30	6	12
画像補正部	30	6	6
色処理部	45	6	4
UCR部	40	3	3.3
ハーフトニング部	80	200	100

各処理部の処理時間がデータサイズに概ね単純比例すると考えれば、表1におけるサンプルJPEGデータが100Kバイトであるのに対し、表2のJPEGデータのサイズは100倍の10Mバイトであるから、処理時間も100倍になる。また、表2に示すデータサイズも単純比例するとして計算したものである。

【0135】次に、表1に示すパフォーマンス値を使って処理部を選択する方法、言い換えれば、処理をデジタルカメラ101およびプリンタ102に分担させる方法を図26を参照して説明する。

【0136】処理分担を決定するデバイスは、ステップS1で、1394ネットワーク103を介して接続されたデバイスから表1に示すような処理能力に関するデータを取得する。つまり、デジタルカメラ101のCPU2401が処理分担を決定する場合は、CPU2401がプリンタ102から処理能

力に関するデータを取得し、プリンタ102のCPU2301が処理分担を決定する場合は、CPU2301がデジタルカメラ101から処理能力に関するデータを取得する。どちらのデバイスが処理分担を決定するかは、第一に、処理分担を決定するためのプログラムがROMに格納されているデバイスであり、第二に、両デバイスのROMにそのようなプログラムが格納されているならば、例えばルートになったデバイスが行うなどすればよい。

【0137】また、一度取得した処理能力に関する情報を不揮発性メモリに格納しておけば、接続されたデバイスが変わらない限り、処理分担を決定する度に処理能力に関するデータを再取得する必要はない。

【0138】次に、処理分担を決定するデバイスは、ステップS2で、下記で説明するような手順によって処理分担を決定する。

【0139】分担を判断する場合、第一に処理時間の短い処理部を選択すると、デジタルカメラ101ではJPEG伸長部251、画像補正部252およびUCR部254が選択され、プリンタ102では画像補正部252、色処理部253およびハーフトーニング部255が選択される。

【0140】画像補正部252のように処理時間がほぼ同じ処理は、直前の処理を行ったデバイス側で行えば、その前処理であるデータ転送を不要にすることができる。従って、画像補正部252はデジタルカメラ101側を選択する方が望ましい。同様に、二つのデバイスで交互に処理を行えば、実際の処理を行う前処理であるデータ転送時間を考慮しなければならない。すなわち、処理を行うデバイスをなるべく切り替えないようにすることが、データ転送時間による処理時間の遅れを最小限にすることになる。

【0141】因みに、1394ネットワークのデータ転送速度は、S400規格であれば40Mバイト/s程度のスピードが得られるが、プロトコルのネゴシエーションなどに必要なオーバーヘッドを加味し実際のスピードは半分程度に低下すると仮定すると、20Mバイト/s程度である。従って、プリンタ102の色処理部253による処理が終了した45MB(表2参照)のCMYデータを、処理が速いデジタルカメラ101のUCR部254へ送るには、二秒以上の転送時間がかかることになる。一方、両デバイスのUCR部254の処理時間の差は0.3秒であるから、プリンタ102の色処理部253を使用するのであれば、そのままプリンタ102のUCR部254で処理を行う方が効率が良い。

【0142】これらの検討結果から、処理時間が最短になるように処理部を選択すると、デジタルカメラ101のJPEG伸長部251→画像補正部252→(データ転送)→プリンタ102の色処理部253→UCR部254→ハーフトーニング部255になり、10MバイトのJPEGデータを処理する合計の処理時間は、 $6+6+1.5+4+3.3+100=120.8$ 秒間と推定される。

【0143】これに対して、デジタルカメラ101ですべての処理を行う場合はハーフトーニング部255の処理後にデータ転送することになるから、10MバイトのJPEGデータを処理する合計の処理時間は、 $6+6+6+3+200+4=225$ 秒間と推定され、 $225-120.8=104.2$ 秒間遅くなると予測される。また、プリンタ102ですべての処理を行う場合はデータ転送後に処理を行うことになるから、10MバイトのJPEGデータを処理する合計の処理時間は、 $0.5+12+6+4+3.3+100=125.8$ 秒間と推定され、 $125.8-120.8=5$ 秒間遅くなると予測される。

【0144】すなわち、処理部を最適に選択した場合、10MバイトのJPEGデータについて、デジタルカメラ101にすべての画像処理を任せる場合より約104秒間早く、プリンタ102にすべての画像処理を任せる場合より約五秒間早く画像処理が終了すると予測される。このような判定結果を基に、デジタルカメラ101またはプリンタ1

02のCPUが一連の画像処理の分担を決定し、その決定に他方のデバイスが従うようにすれば、デジタルカメラとプリンタとを組み合わせでダイレクトプリントを行おうとする場合、最も処理時間を短くすることができる画像処理の分担を適応的に設定することが可能になる。

【0145】上記のようにして処理分担を決定したデバイスは、ステップS3で、1394ネットワーク103を介して相手のデバイスに処理分担を通知する。この通知を受信したデバイスは、通知された処理分担に従い、自身が行う処理および送信または受信するデータの形態を決定する。

【0146】また、一度決定された処理分担を不揮発性メモリに格納しておけば、接続されたデバイスが変わらない限り、処理分担を再決定する必要はない。

【0147】このように、本実施形態によれば、1394シリアルバス103で接続されたデジタルカメラ101などの画像供給デバイスおよびプリンタ102などの印刷デバイス間でダイレクトプリントを行う際に、それぞれのデバイスがもつ画像処理機能ごとの処理能力、並びに、データサイズおよびデータ転送速度の関係に応じて適応的に画像処理をデバイス間で分担することができるので、効率が良いダイレクトプリントを行うことが可能になる。

【0148】

【他の実施形態】本発明は、複数の機器(例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど)から構成されるシステムに適用しても、一つの機器からなる装置(例えば、複写機、ファクシミリ装置など)に適用してもよい。

【0149】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体(または記録媒体)を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ(またはCPUやMPU)が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることはいうまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【0150】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示

に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【0151】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードが格納されることになる。

【0152】

【発明の効果】以上説明したように、本発明によれば、個々のデバイスに画像処理を分担することで効率的な画像処理を行うことができる。

【図面の簡単な説明】

【図1】本発明を適用するシステムの一般的な構成例を示す図、

【図2】1394インタフェースを備えるノードにより構成される1394ネットワークの構成例を示す図、

【図3】1394インタフェースの構成要素を説明する図、

【図4】リンクレイヤが提供可能なサービスを示す図、

【図5】トランザクションレイヤが提供可能なサービスを示す図、

【図6】1394インタフェースにおけるアドレス空間を説明する図、

【図7】CSRコアレジスタに格納される情報のアドレスおよび機能を示す図、

【図8】シリアルバスレジスタに格納される情報のアドレスおよび機能を示す図、

【図9】最小形式のコンフィグレーションROMの構成を示す図、

【図10】一般形式のコンフィグレーションROMの構成を示す図、

【図11】ユニット空間のシリアルバス装置レジスタに格納される情報のアドレスおよび機能を示す図、

*【図12】IEEE1394規格に準拠した通信ケーブルの断面図、

【図13】DS-Link符号化方式を説明する図、

【図14】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図15】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図16】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図17A】各ノードにノードIDを割り当てる処理を詳細に示すフローチャート、

【図17B】各ノードにノードIDを割り当てる処理を詳細に示すフローチャート、

【図18】セルフIDパケットの構成例を示す図、

【図19】図2に示したネットワーク構成におけるアービトレーションを説明する図、

【図20】一通信サイクルにアイソクロナス転送モードおよびアシンクロナス転送モードを混在させた状態を説明する図、

20 【図21】アイソクロナス転送されるパケットのフォーマットを示す図、

【図22】アシンクロナス転送されるパケットのフォーマットを示す図、

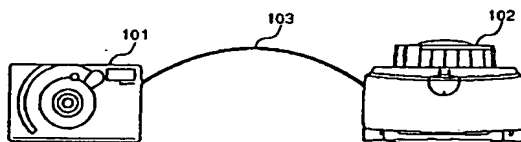
【図23】図1に示すプリンタの内部構成例を示すブロック図、

【図24】図1に示すデジタルカメラの内部構成例を示すブロック図、

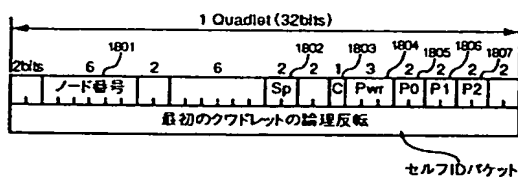
30 【図25】デジタルカメラで撮影されたJPEG圧縮された画像データを、プリンタに合わせた印刷データに変換する処理を示すブロック図、

【図26】画像処理をデジタルカメラおよびプリンタに分担させる方法を説明するフローチャートである。

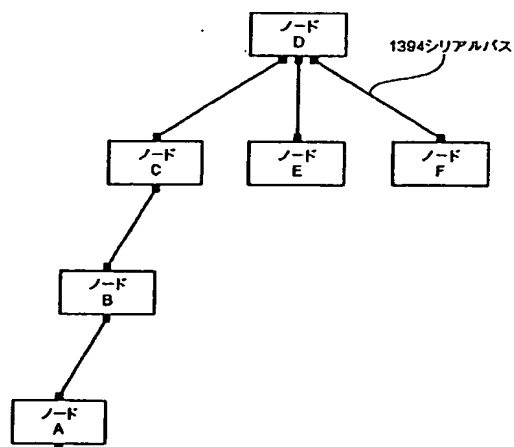
【図1】



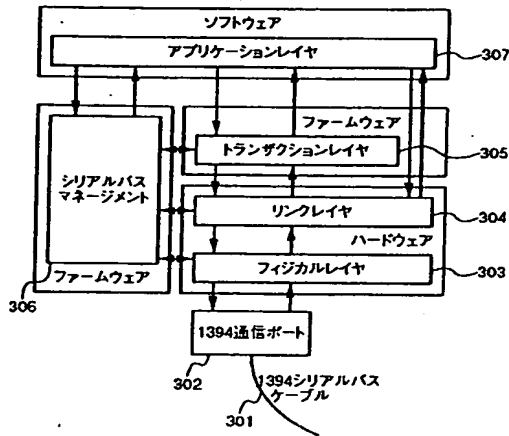
【図18】



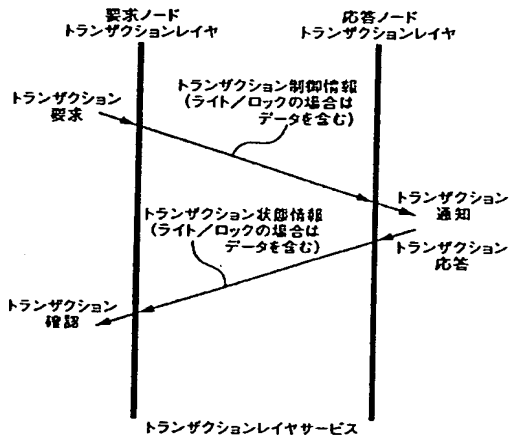
【図2】



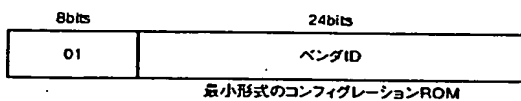
【図3】



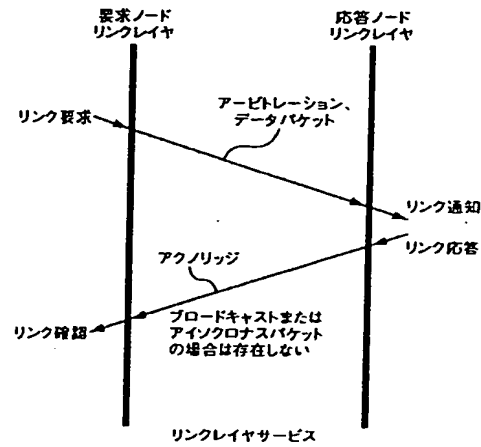
【図5】



【図9】



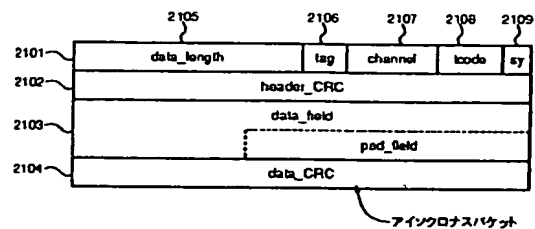
【図4】



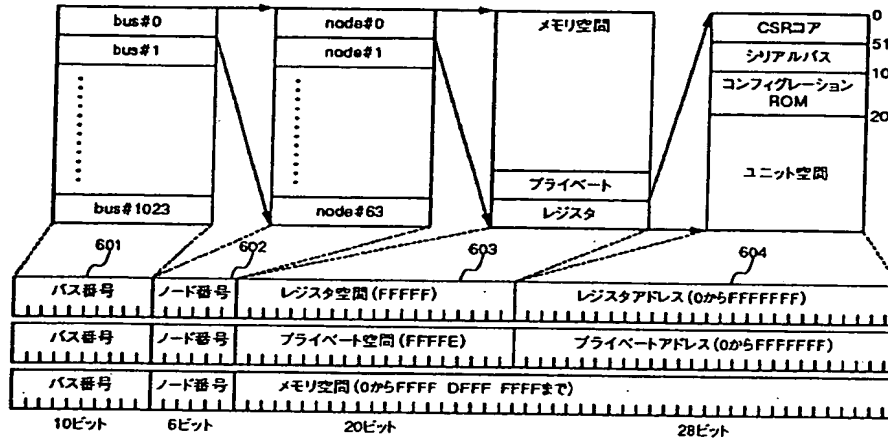
【図7】

CRSコアレジスタ		
オフセット (16進数)	レジスタ名称	機能
000	STATE_CLEAR	状態と制御の情報
004	STATE_SET	STATE_CLEARの書き込み可否を示す情報
008	NODE_IDS	バスID+ノードID
00C	RESET_START	この領域に対する書き込みでバスをリセットする
010~014	INDIRECT_ADDRESS, INDIRECT_DATA	1Kより大きいROMをアクセスするためのレジスタ
018~01C	SPLIT_TIMEOUT	スプリットトランザクションのタイムアウトを検出するタイマの値
020~02C	ARGUMENT, TEST_START, TEST_STATUS	診断用のレジスタ
030~04C	UNITS_BASE, UNITS_BOUND, MEMORY_BASE, MEMORY_BOUND	IEEE1394では実装しない
050~054	INTERRUPT_TARGET, INTERRUPT_MASK	割り込み通知レジスタ
058~07C	CLOCK_VALUE, CLOCK_TICK_PERIOD, CLOCK_STROBE_ARRIVED, CLOCK_INFO	IEEE1394では実装しない
080~0FC	MESSAGE_REQUEST, MESSAGE_RESPONSE	メッセージ通知レジスタ
100~17C		予約
180~1FC	ERROR_LOG_BUFFER	IEEE1394用に予約

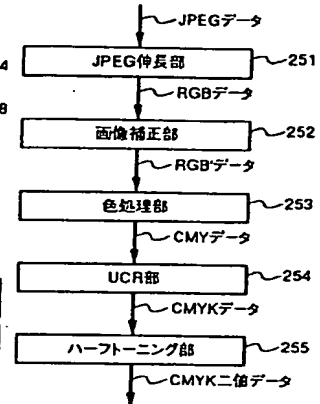
【図21】



【図6】



【図25】



【図8】

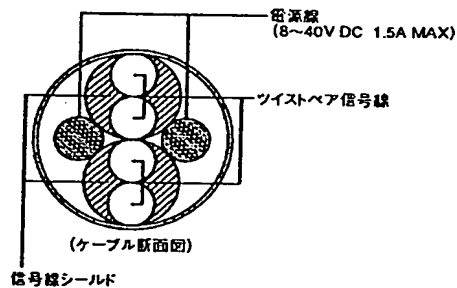
シリアルバスレジスタ		
オフセット (16進数)	レジスタ名称	機能
200	CYCLE_TIME	アイソクロナス転送のためのカウンタ
204	BUS_TIME	時間を同調するためのレジスタ
208	POWER_FAIL_IMMINENT	電源供給に関するレジスタ
20C	POWER_SOURCE	
210	BUSY_TIMEOUT	トランザクションレイヤの再試行を制御する
214 ↓ 218		予約
21C	BUS_MANAGER_ID	バスマネージャのノードID
220	BANDWIDTH_AVAILABLE	アイソクロナス転送の帯域を管理する
224 ↓ 228	CHANNELS_AVAILABLE	アイソクロナス転送のチャネル番号を管理する
22C	MAINT_CONTROL	診断用レジスタ
230	MAINT_UTILITY	
234 ↓ 3FC		予約

【図10】

Bus Info Block Length	ROM Length	CRC
Bus Info Block		
Root Directory		
Node Dependent Info Directory		
Unit Directories		
Root & Unit Leaves		
Vendor Dependent Information		

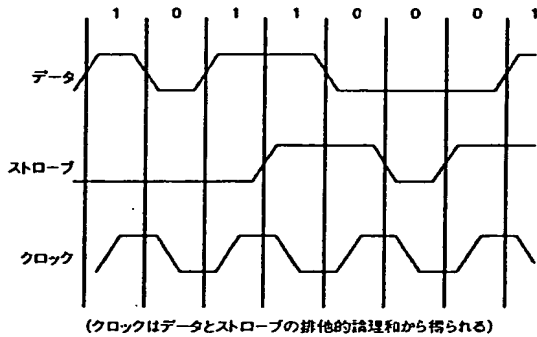
【図11】

【図12】

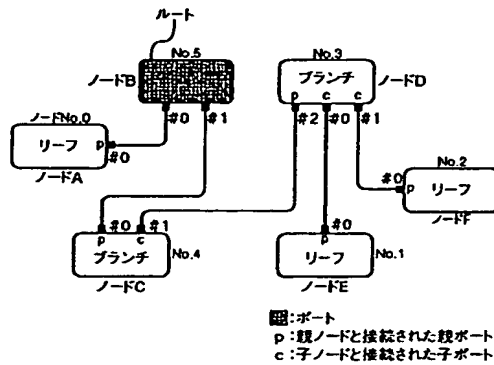


シリアルバス設置レジスタ		
オフセット (16進数)	レジスタ名称	機能
800 ↓ FFC		予約
1000 ↓ 13FC	TOPOLOGY_MAP	シリアルバスの構成情報
1400 ↓ 1FFC		予約
2000 ↓ 2FFC	SPEED_MAP	シリアルバスの転送速度の情報
3000 ↓ FFFC		予約

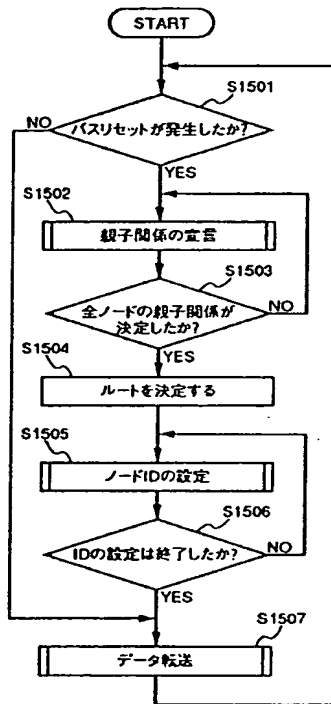
【図13】



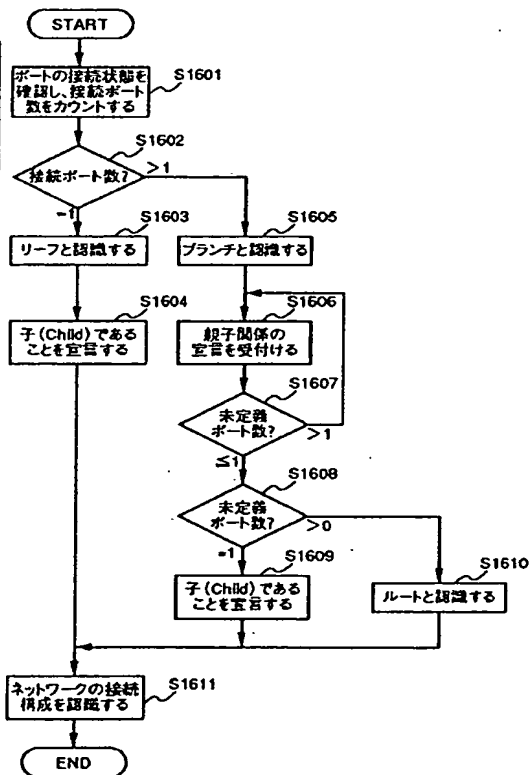
【図14】



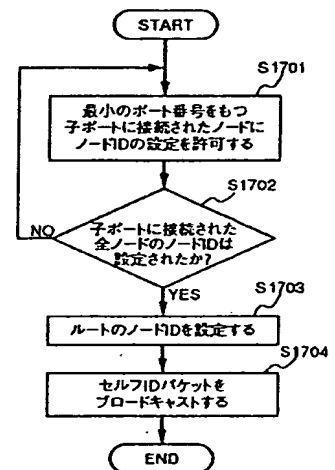
【図15】



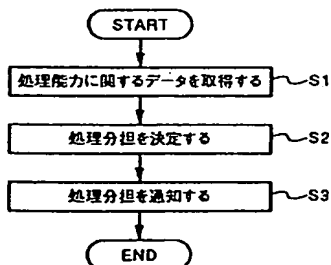
【図16】



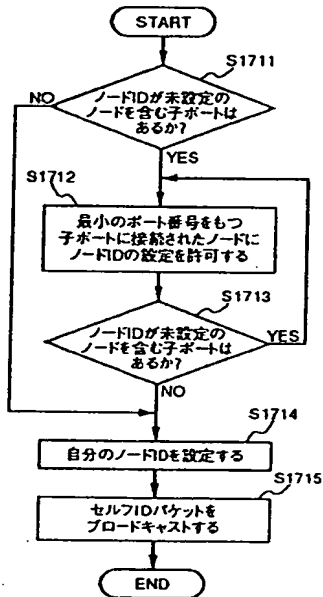
【図17A】



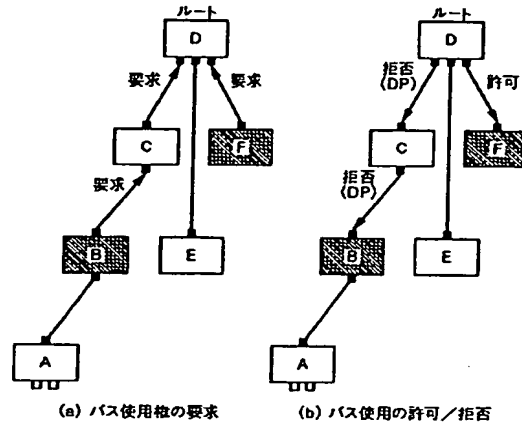
【図26】



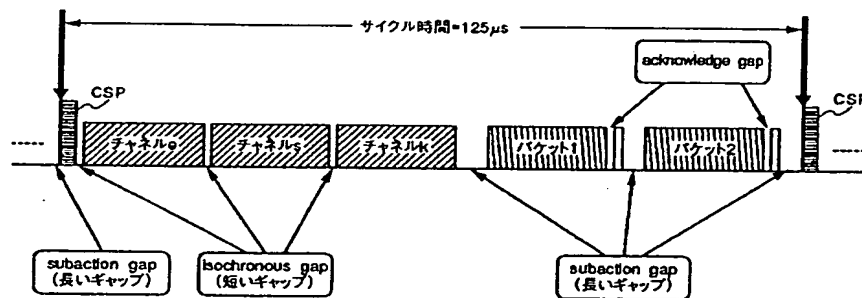
【図17B】



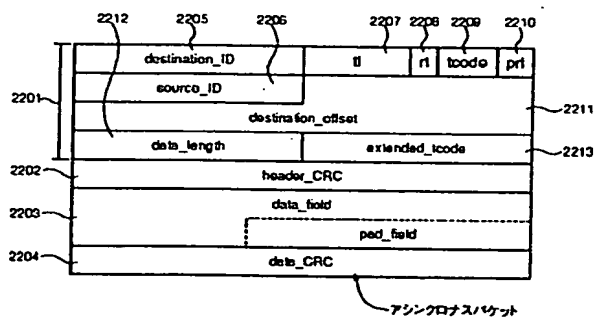
【図19】



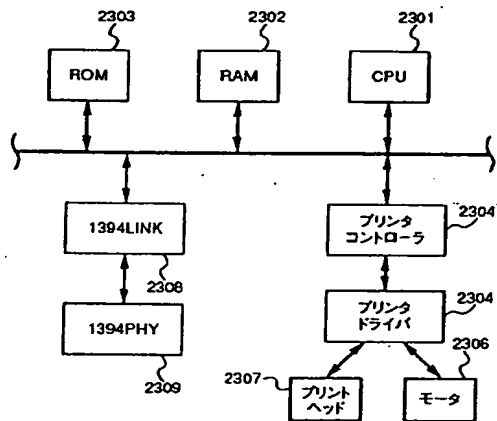
【図20】



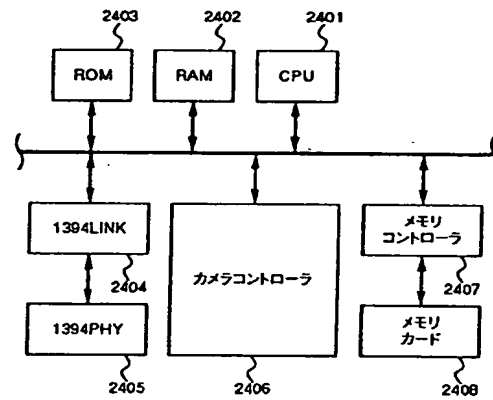
【図22】



【図23】



【図24】



フロントページの続き

(51)Int.Cl.

識別記号

F I

テーマコード (参考)

// H 0 4 N 101:00

F ターム (参考) SC022 AA01 AA11 AA13 AB01 AB21
AC03 AC42 AC75
SC052 AA11 AB02 CC06 CC11 DD02
DD04 FA02 FA03 FA07 FB01
FC06 FE04
SC064 BA04 B805 BC10 BC16 BC25
BD02 BD08

(11) Japanese Patent Application

Laid-open (KOKAI) No. 2001-275066

(43) Laid-opened Date: October 5, 2001

(54) Title of the invention:

5 IMAGE PROCESSING APPARATUS, METHOD THEREFOR
 AND STORAGE MEDIUM

(21) Application Number: 2000-89304

(22) Filing Date: March 28, 2000

(71) Applicant: CANON KABUSHIKI KAISHA

10 (72) Inventor: Jiro TATEYAMA

(54) [Title of the Invention]

IMAGE PROCESSING APPARATUS, METHOD THEREFOR AND STORAGE
15 MEDIUM

(57)[Abstract]

[Problems]

 There is no determination criterion for
20 determining which processing should be performed by
which device in order to perform efficient printing, in
a case where each of devices supporting direct print
has a series of processing functions.

[Solving means]

25 When direct print is performed between an image
supply device such as a digital camera 101 and a

printing device such as a printer 102 connected via a
1394 serial bus 103, image processing is appropriately
shared by the devices based on the processing capacity
of each image processing function of each device and
5 relation between the data size and the data transfer
rate to perform efficient direct print.

[Claims for the Patent]

[Claim 1]

An image processing apparatus for supplying data
10 into a printing apparatus connected via a serial bus,
characterized by comprising:

image processing means for converting image data
to print data; and

decision means for deciding sharing of image
15 processing with the printing apparatus.

[Claim 2]

An image processing apparatus for receiving data
from a data supply apparatus connected via a serial bus,
characterized by comprising:

20 image processing means for converting image data
into print data; and

decision means for deciding sharing of image
processing with the data supply apparatus.

[Claim 3]

25 The image processing apparatus according to claim
1 or 2, characterized by further comprising acquisition
means for acquiring capacity information of image

processing from a counterpart apparatus.

[Claim 4]

The image processing apparatus according to claim
3, characterized in that the capacity information is
5 provided for each step of image processing.

[Claim 5]

The image processing apparatus according to claim
3 or 4, characterized in that said decision means
decides sharing of image processing based on the
10 capability information, image data size and data
transfer rate.

[Claim 6]

The image processing apparatus according to claim
5, said decision means decides sharing of image
15 processing so that data transfer efficiency is enhanced.

[Claim 7]

The image processing apparatus according to any
one of claims 1 to 6, characterized by further
comprising notification means for notifying a
20 counterpart apparatus of the decided sharing of image
processing.

[Claim 8]

The image processing apparatus according to any
one of claims 1 to 7, characterized in that the
25 apparatus which supplies image data is an image
acquisition device, and the apparatus which is supplied
with the data is a printer.

[Claim 9]

The image processing apparatus according to any one of claims 1 to 8, characterized in that the serial bus is in accordance with or in conformity with the
5 IEEE 1394 standard.

[Claim 10]

The image processing apparatus according to claim 9, characterized in that the capacity information is determined by reading information of configuration ROM
10 specified in conformity with the IEEE 1394 standard.

[Claim 11]

The image processing apparatus according to any one of claims 1 to 8, characterized in that the serial bus is in accordance with or in conformity with the USB
15 standard.

[Claim 12]

An image processing method for supplying data to a printing apparatus connected via a serial bus, characterized in that:

20 steps of image processing for converting image data to print data are shared with the printing apparatus.

[Claim 13]

An image processing method for being supplied with
25 data from a data supply apparatus connected via a serial bus, characterized in that:

steps of image processing for converting image

data to print data are shared with the data supply apparatus.

[Claim 14]

The image processing method according to claim 12
5 or 13, characterized by further comprising a step of
obtaining the capability information of image
processing from a counterpart apparatus.

[Claim 15]

The image processing method according to claim 14,
10 characterized in that the capacity information is
provided for each step of image processing.

[Claim 16]

The image processing method according to claim 14
or 15, characterized in that the sharing is decided
15 based on the capability information, image data size
and data transfer rate.

[Claim 17]

The image processing method according to claim 16,
characterized in that the sharing is decided so that
20 transfer is performed at a time.

[Claim 18]

The image processing method according to any one
of claims 12 to 17, characterized by further comprising
a step of notifying the decided sharing of image
25 processing to a counterpart apparatus.

[Claim 19]

The image processing method according to any one

of claims 12 to 18, characterized in that the apparatus which supplies image data is an image acquisition device, and the apparatus which is supplied with the data is a printer.

5 [Claim 20]

The image processing method according to any of claims 12 to 19, characterized in that the serial bus is in accordance with or in conformity with the IEEE 1394 standard.

10 [Claim 21]

The image processing method according to claim 20, characterized in that the capacity information is determined by reading information of configuration ROM specified in conformity with the IEEE 1394 standard.

15 [Claim 22]

The image processing method according to any of claims 12 to 19, characterized in that the serial bus is in accordance with or in conformity with the USB standard.

20 [Claim 23]

A storage medium in which a program code for image processing for supplying data to a printing apparatus connected via a serial bus is stored, characterized in that

25 the program code at least has a code for a step of sharing steps of image processing for converting image data to print data with the printing apparatus.

[Claim 24]

A storage medium in which a program code for image processing for being supplied with data from a data supply apparatus connected via a serial bus,
5 characterized in having:

a code for a step of sharing steps of image processing for converting image data to print data with the data supply apparatus.

[Detailed Description of the Invention]

10 [0001]

[Field of the Invention]

The present invention relates to an image processing apparatus, a method therefor and a storage medium and, for example, to an image processing
15 apparatus between devices connected via a serial bus and a method therefor.

[0002]

[Prior Art]

In order to print an image taken by a digital
20 camera, a procedure is required in which image data is transferred to a personal computer via a serial interface such as RS-232C and a memory card, processing suitable for the printing format of a printer is performed for the image data by the personal computer,
25 and then print data is sent to the printer via a parallel interface such as Centronics or a serial interface such as USB (Universal Serial Bus) and

printed.

[0003]

If a user of the digital camera already has a personal computer, a common system configuration is used in which image processing is performed with the use of necessary application software, and printing is performed by a printer connected to the personal computer. However, in the case of a user who has purchased a digital camera but does not have a personal computer, the only way in which he can use the digital camera is to connect the digital camera to the video terminal of a home TV set and view taken images on the TV screen because he does not have means for printing the taken images.

15 [0004]

There exists a printer system in the form of a video printer for such users. In this printer system, image data is transferred directly from a digital camera to a printer not via a personal computer but via an original-standard serial interface, an infrared interface or a memory card, and printing is performed after image processing is performed inside the printer. To transfer image data not via a personal computer but directly to a printer for printing is referred to as "direct print".

25 [0005]

[Problems to be Solved by the Invention]

In a print system which performs direct print, it is necessary for a device (a digital camera or a printer) to perform processing for converting JPEG (Joint Photographic Experts Group) compressed image data to print data. Accordingly, time required for printing significantly depends on the data processing capacity of the device.

[0006]

For example, in a common general-purpose printer, it is assumed that processing for converting image data to print data specific to the printer is performed by a personal computer, and therefore, specifications for the printer are prepared so that priority is given to cost without providing high data processing capacity for the printer.

[0007]

A printer which supports direct print is required to expand its capabilities, such as speed-up of the CPU mounted on the printer, distributed processing with the use of multiple CPU's, increase in capacity of an internal memory to increase the size of data which can be converted at once, and the like in order to enhance the processing capacity for converting image data to print data.

[0008]

It is also necessary to provide an image supply device (for example, a digital camera) which supports

direct print with a print data creation function.

However, a general-purpose digital camera and the like have a sufficiently high-level capability in image data compression, expansion and display functions but do not
5 have a sufficiently high-level print data creation capability.

[0009]

Furthermore, there is a problem that, at present, there is no determination criterion for determining
10 which processing should be performed by which device to perform efficient printing, in a case where each of devices supporting direct print has a series of processing functions.

[0010]

15 The present invention has been made to solve the above problems, and its object is to perform efficient image processing by causing individual devices to share the image processing.

[0011]

20 [Means for Solving the Problems]

As means for achieving the above-described object, the present invention has the following configuration.

[0012]

An image processing apparatus according to the
25 present invention is an image processing apparatus for supplying data to a printing apparatus connected via a serial bus, characterized in comprising: image

processing means for converting image data to print data; and decision means for deciding sharing of image processing with the printing apparatus.

[0013]

5 Furthermore, an image processing apparatus according to the present invention is an image processing apparatus which is supplied with data from a data supply apparatus connected via a serial bus, characterized in comprising: image processing means for
10 converting image data to print data; and decision means for deciding sharing of image processing with the data supply apparatus.

[0014]

 An image processing method according to the
15 present invention is an image processing method for supplying data to a printing apparatus connected via a serial bus, characterized in that steps of image processing for converting image data to print data are shared with the printing apparatus.

20 [0015]

 Furthermore, an image processing method according to the present invention is an image processing method for being supplied with data from a data supply apparatus connected via a serial bus, characterized in
25 that steps of image processing for converting image data to print data are shared with the data supply apparatus.

[0016]

[Embodiments of the Invention]

A data transfer method of an embodiment according to the present invention will be described in detail
5 with reference to drawings.

[0017]

Figure 1 shows an example of common configuration of a system to which the present invention is applied, and shows a direct print system in which a digital
10 camera 101 and a printer 102 are connected via serial bus in accordance with the IEEE 1394-1995 standard (hereinafter referred to as a "1394 serial bus"). The outline of the IEEE 1394-1995 standard (hereinafter referred to as the "IEEE 1394 standard") will be
15 described first.

[0018]

The details of the IEEE 1394 standard are described in "IEEE Standard for a High Performance Serial Bus" published by IEEE (The Institute of
20 Electrical and Electronics Engineers, Inc.) on August 30, 1996.

[0019]

Connection between the digital camera 101 and the printer 102 is not limited to a 1394 serial bus. They
25 may be connected via a USB serial bus.

[0020]

Outline

Figure 2 shows an example of configuration of a communication system (hereinafter referred to as "a 1394 network") which is configured by nodes provided with a digital interface in conformity with the IEEE 1394 standard (hereinafter referred to as "a 1394 interface"). The 1394 network forms a bus topology network enabling serial data communication.

[0021]

In Figure 2, nodes A to H are connected via communication cables in conformity with the IEEE 1394 standard. These nodes A to H are, for example, electronic equipment such as a PC (personal computer), a digital VTR (video tape recorder), a DVD (digital video disc) player, a digital camera, a hard disk and a monitor.

[0022]

The connection method for the 1394 network corresponds to a daisy chain method or a node branch method enabling connection with high flexibility.

[0023]

In the 1394 network, for example, bus reset is automatically performed if existing equipment is separated from the network, new equipment is added to the network, or existing equipment is powered on/off. This bus reset enables the 1394 network to automatically recognize new network connection configuration and assign ID information to each

equipment. In other words, this function makes it possible for the 1394 network to always recognize network connection configuration.

[0024]

5 Furthermore, the 1394 network has a function of relaying data transferred from other equipment, and this function makes it possible for all the equipment to grasp the operation condition of the 1394 serial bus.

[0025]

10 Furthermore, the 1394 network has a function called Plug&Play, and this function makes it possible, only by connecting equipment to the network without powering off all the equipment connected to the network, to cause the connected equipment to be automatically
15 recognized.

[0026]

 The 1394 network supports the data transfer rates of 100, 200 and 400 Mbps. Since equipment with a higher data transfer rate can support a lower data
20 transfer rate, equipment supporting different data transfer rates can be connected with one another.

[0027]

 Furthermore, the 1394 network supports two different data transfer methods, that is, an
25 asynchronous transfer mode and an isochronous transfer mode. The asynchronous transfer mode is effective in transferring data required to be asynchronously

transferred as necessary, that is, a control signal or a data file. The isochronous transfer mode is effective in transferring such data that a predetermined amount of the data is required to be continuously transferred at a constant data rate, that is, video data, audio data or the like.

[0028]

It is possible for the asynchronous transfer mode and the isochronous transfer mode to co-exist in each communication cycle (commonly, 1 cycle is 125 μ s). Each transfer mode is implemented after a cycle start packet (CSP) indicating start of a cycle is transferred. In each communication cycle period, higher priority is set for the isochronous transfer mode than for the asynchronous transfer mode. The transfer band for the isochronous transfer mode is secured in each communication cycle.

[0029]

[Architecture]

Figure 3 illustrates components of a 1394 interface.

[0030]

The 1394 interface is functionally configured by multiple layers. The 1394 interface is connected to 1394 interfaces of another node via a communication cable 301 which is in conformity with the IEEE 1394 standard. Furthermore, the 1394 interface has one or

more communication ports 302, and the communication port 302 is connected to a physical layer 303 included in hardware.

[0031]

5 The hardware is configured by the physical layer 303 and a link layer 304. The physical layer 303 performs physical or electrical interfacing with other nodes, detection of bus reset and accompanying processing, coding/decoding of input/output signals, arbitration for the bus use right and the like. The link layer 304 performs generation and sending/receiving of a communication packet, control of a cycle timer and the like.

[0032]

15 In Figure 3, firmware includes a transaction layer 305 and a serial bus management 306. The transaction layer 305 manages the asynchronous transfer mode and provides various transactions (read, write and lock). The serial bus management 306 provides functions of controlling the node, managing the connection condition of and ID information about the node and performing resource management of the 1394 network based on CSR architecture to be described later.

[0033]

25 The above-described hardware and the firmware substantially constitute the 1394 interface, and the basic configuration is specified by the IEEE 1394

standard.

[0034]

An application layer 307 included in software differs according to application software to be used, and it controls how data should be communicated over the 1394 network. For example, a communication protocol such as AV/C Protocol is specified for motion images of digital VTR.

[0035]

10 • Link layer

Figure 4 shows services which can be provided by the link layer 304.

[0036]

The link layer 304 provides the following four services. A link response (LK_DATA.response) does not exist in the case of broadcast communication or transfer of an isochronous packet.

(1) Link request (LK_DATA.request): Requests a responding node to transfer a predetermined packet.

20 (2) Link notification (LK_DATA.indication):
Notifies a responding node of receiving of a predetermined packet.

(3) Link response (LK_DATA.response): Sends acknowledge from a responding node.

25 (4) Link confirmation (LK_DATA.confirmation):
Notifies a requesting node of receiving of acknowledge.

[0037]

The link layer 304 realizes the above-stated two kinds of transfer mode, that is, the asynchronous transfer mode and the isochronous transfer mode based on the above-described services.

5 [0038]

- Transaction layer

Figure 5 shows services which can be provided by the transaction layer 305.

[0039]

10 The transaction layer 305 provides the following four services.

(1) Transaction request (TR_DATA.request):

Requests a predetermined transaction from a responding node.

15 (2) Transaction notification (TR_DATA.indication):

Notifies a responding node of receiving of a predetermined transaction request.

(3) Transaction response (TR_DATA.response):

Transaction state information from a responding node

20 (including data in the case of write/lock).

(4) Transaction confirmation

(TR_DATA.confirmation): Notifies a requesting node of receiving of transaction state information.

[0040]

25 The transaction layer 305 manages asynchronous transfer based on the above-described services, and realizes the following three kinds of transaction, that

is, read transaction, write transaction and lock transaction.

(1) Read transaction: A requesting node reads information stored at a particular address of a responding node.

(2) Write transaction: A requesting node writes predetermined information at a particular address of a responding node.

(3) Lock transaction: Reference data and update data are transferred from a requesting node to a responding node; information at a particular address of the responding node and the reference data are compared; and the information at the particular address is overwritten with the update data based on the comparison result.

[0041]

- Serial bus management

The serial bus management 306 provides the following three functions.

(1) Node control: Provides a function of managing each of the above-described layers and managing asynchronous transfer performed with other nodes.

(2) Isochronous resource manager (IRM): Provides a function of managing isochronous transfer performed with other nodes.

(3) Bus manager: Provided with the IRM function and provides a higher-level bus management function

than IRM.

[0042]

Specifically, the IRM manages information required
for assignment of transfer band width and channel
5 numbers and provides the information to other nodes.
There exists only one IRM on a local bus, and it is
dynamically selected from among other candidates (nodes
provided with the IRM function) every time bus reset is
performed. The IRM may provide a part of functions
10 which can be provided by the bus manager (management of
connection configuration, power source management,
management of rate information and the like).

[0043]

Specifically, the bus manager performs higher-
15 level management and optimization of the 1394 serial
bus based on the management information, and it further
has a function of providing the information to other
nodes. The higher-level management means management of
information about whether power can be supplied via a
20 communication cable or not, whether power source supply
is necessary or not, and the like for each node (high-
level power source management), management of the
maximum transfer rate among nodes (high-level rate
information management), and creation of a topology map
25 (high-level connection configuration management).

[0044]

The bus manager can provide services for

controlling the 1394 network for an application. The services include the following three services.

(1) Serial bus control request

(SB_CONTROL.request): A service that an application
5 requests bus reset.

(2) Serial bus event control confirmation

(SB_CONTROL.confirmation): A service that an application is confirmed about a serial bus control request.

10 (3) Serial bus event notification

(SB_CONTROL.indication): A service that an application is notified of an event that asynchronously occurs.

[0045]

[Address specification]

15 Figure 6 illustrates address space in the 1394 interface. In the 1394 interface, a 64-bit width address space is specified in accordance with a CSR (Command and Status Register) architecture in conformity with the ISO/IEC 13213:1994.

20 [0046]

In Figure 6, the first 10-bit field 601 is used for a number which specifies a predetermined 1394 serial bus, and the next 6-bit field 602 is used for a number which specifies predetermined equipment (a node).

25 These higher order 16 bits are called "a node ID", and each node identifies another node with this node ID. Furthermore, each node can use this node ID to identify

a counterpart to perform communication.

[0047]

The field constituted by the remaining 48 bits corresponds to address space (256 MB structure) provided for each node, and a 20-bit field 603 among the 48 bits specifies multiple areas constituting the address space. The area from "0 to 0xFFFFD" in the field 603 is called a memory space, and the area of "0xFFFFE" is called a private space, which can be freely used by each node. The area of "0xFFFF" is called a register space, in which information common to nodes connected to the bus is stored. Each node can manage communication among nodes with the use of the information stored in the register space.

15 [0048]

The last 28-bit field 604 specifies an address at which common information or information specific to each node is stored. For example, in the register space, the first 512 bytes are used as a CSR architecture core (CSR core) register. Addresses and functions of information stored in the CSR core register are shown in Figure 7. The offset values shown in Figure 7 indicate relative locations from "0xFFFFF0000000".

25 [0049]

The subsequent 512 bytes are used as a register for a serial bus. Addresses and functions of

information stored in the serial bus register are shown in Figure 8. The offset values shown in Figure 8 indicate relative locations from "0xFFFFF0000200".
[0050]

5 The subsequent 1024 bytes are used as a configuration ROM. There are a minimum form and a general form for the configuration ROM arranged from "0xFFFFF0000400". The configuration of a minimum-form configuration ROM is shown in Figure 9. In Figure 9, a
10 vender ID is a numeric value of 24 bits which is assigned specifically to each vender. The configuration of a general-form configuration ROM is shown in Figure 10. In Figure 10, the above-described vender ID is stored in Root Directory 1002. In Bus
15 Info Block 1001 and Root Leaf 1005, a node unique ID can be held as specific ID information which identifies each node.
[0051]

20 It is specified that a specific ID which can identify one device irrespective of the manufacturers or the model should be determined as the node unique ID. A node unique ID is constituted by 64 bits. The higher order 24 bits indicate the above-described vendor ID, and the lower order 48 bits indicate information which
25 can be freely set by the manufacturer of each device (for example, a serial number of the device). This node unique ID is used, for example, to continuously

recognize a particular device before and after bus reset.

[0052]

In Root Directory 1002 shown in Figure 10,
5 information about basic functions of the device can be held. Detailed function information is stored in subdirectories, Unit Directories 1004, which are offset from Root Directory 1002. In Unit Directories 1004, for example, information about a software unit
10 supported by the device is stored. Specifically, information about data transfer protocols for performing data communication among nodes and a command set for defining a predetermined communication procedure are held.

15 [0053]

In Node Dependent Info Directory 1003 shown in Figure 10, device-specific information can be held. Node Dependent Info Directory 1003 is offset by Root Directory 1002.

20 [0054]

Furthermore, in Vendor Dependent Information 1006 shown in Figure 10, information specific to a vendor of manufacturing or selling the node can be held.

[0055]

25 The remaining area of the field 604 shown in Figure 6 is called a unit space, which specifies an address at which information specific to each device,

for example, identification information (a company name, a model name or the like) of each equipment or use conditions are stored. The addresses and functions of information stored in a serial bus apparatus register in the unit space are shown in Figure 11. The offset values shown in Figure 11 indicate relative locations from "0xFFFFF0000800".

[0056]

In general, in order to simplify design of a different kind of bus system, each device should use only the first 2048 bytes of the unit space. That is, it is desirable to make configuration by 4096 bytes obtained by adding 2048 bytes of the CSR core register, the serial bus register and the configuration ROM and the first 2048 bytes of the unit space.

[0057]

[Communication cable]

Figure 12 is a sectional view of a communication cable in conformity with the IEEE 1394 standard.

[0058]

The communication cable is configured by two pairs of twisted-pair signal lines and power source lines. By providing power source lines for the communication cable, the 1394 interface can supply power even to equipment the main power source of which is off or equipment the power of which has been dropped due to a failure. The DC power which can be supplied via the

power-source lines is specified to be 8 to 40 V and up to 1.5 A.

[0059]

The two pairs of twisted-pair signal lines
5 transmit an information signal coded by the DS-Link
(Data/Strobe Link) coding method. Figure 13
illustrates the DS-Link coding method.

[0060]

The DS-Link coding method is suitable for high-
10 speed serial data communication, and the configuration
requires two pairs of twisted-pair signal lines. One
pair of signal lines sends a data signal, and the other
pair of signal lines sends a strobe signal. The
receiving side can generate a clock by performing
15 exclusive-OR of the data signal and the strobe signal
received via the two pairs of signal lines. A 1394
interface using the DS-Link coding method has, for
example, the following advantages:

(1) The transfer efficiency is higher in
20 comparison with other coding methods.

(2) Since necessity of a PLL (Phase Locked Loop)
circuit is eliminated, the circuit scale of a
controller LSI can be reduced.

(3) Since necessity of sending a signal indicating
25 an idle state is eliminated, a transceiver circuit can
be easily switched to a sleep state, and thereby power
consumption can be reduced.

[0061]

[Bus reset]

Each node (strictly, the 1394 interface of each node) can automatically detect that the network
5 connection configuration has been changed. In this case, the 1394 network performs processing called bus reset in accordance with the procedure shown below. The change in the connection configuration is detected by change in bias voltage applied to a communication
10 port provided for each node.

[0062]

A node which has detected change in the network connection configuration, for example, increase/decrease in the number of nodes due to
15 connection/disconnection of a node, on/off of the power source of a node or the like, or a node which requires recognition of new connection configuration sends a bus reset signal to the 1394 serial bus via the 1394 interface.

20 [0063]

A node which has received the bus reset signal communicates generation of bus reset to its link layer 304 and transfers the bus reset signal to another node. The node which has received the bus reset signal clears
25 the network connection configuration which it has recognized and the node ID which has been assigned to each equipment. After all the nodes eventually detect

the bus reset signal, each node automatically performs initialization processing accompanying the bus reset, that is, recognition of new connection configuration and assignment of a new node ID.

5 [0064]

In addition to the above-described case where bus reset is activated by change in connection configuration, bus reset may also be activated by the application layer 307 issuing an instruction directly
10 to the physical layer 303 when initiated by a host. When bus reset is activated, data transfer is suspended. After the initialization processing accompanying the bus reset is finished, the data transfer is restarted under new network connection configuration.

15 [0065]

- Bus reset sequence

As described above, after bus reset is activated, each node automatically performs recognition of new connection configuration and assignment of a new node
20 ID. A basic sequence from start of bus reset to assignment of a node ID will be described with the use of Figures 14 to 16.

[0066]

Figure 14 illustrates the condition after bus
25 reset is activated in the 1394 network shown in Figure 2. In Figure 14, nodes A, E and F have one communication port; nodes B and C have two

communication ports; and a node D has three communication ports. To each communication port, there is given a port number for identifying the port.

[0067]

- 5 Description will be made below on a process from start of bus reset to assignment of a node ID in the network connection configuration shown in Figure 14, with the use of a flowchart shown in Figure 15.

[0068]

- 10 At step S1501, each of the nodes A to F constituting the 1394 network are continuously monitoring generation of bus reset, and when a bus reset signal is outputted from a node which has detected change in connection configuration, each node
- 15 performs the following processing.

[0069]

- When bus reset occurs, respective nodes declare parent-child relationship between communication ports provided for the nodes, at step S1502. The parent-
- 20 child relationship declaration is repeated until it is determined at step S1503 that the parent-child relationship has been determined among all the ports.

[0070]

- When the parent-child relationship has been
- 25 determined among all the ports, a node for performing network arbitration, that is, a root is determined at step S1504. Then, at step S1505, the 1394 interface of

each node performs a work for automatically setting a node ID for the node. The setting of a node ID is repeated until it is determined that node ID's have been set for all the nodes.

5 [0071]

When the node ID's have been set for all the nodes, each node performs data transfer through isochronous transfer or asynchronous transfer at step S1507. In Figure 15, it is described that that the processing of
10 step S1501 is performed after the processing of step S1507 is finished. However, strictly, each node monitors generation of bus reset at step S1501 while performing data transfer of step S1507. Then, if bus reset occurs, each node stops data transfer and, after
15 performing processing from step S1502 to S1506, restarts data transfer under new connection configuration.

[0072]

In accordance with the above-described procedure,
20 the 1394 interface of each node can automatically perform recognition of new connection configuration and assignment of a new node ID every time bus reset occurs.

[0073]

• Decision of parent-child relationship

25 Figure 16 is a flowchart showing a process for declaration of parent-child relationship at step S1502, that is, recognition of parent-child relationship

between ports in detail.

[0074]

At step S1601, each node checks the connection condition of communication ports provided for the node, that is, whether the ports are connected or
5 disconnected, and counts the number of communication ports connected to other nodes (hereinafter referred to as "connected ports"). Then, at step S1602, the number of connected ports is determined, and a node with the
10 number of connected ports of "1" recognizes itself as a "leaf" at step S1603. A leaf is a node which is connected to only one different node. In Figure 14, the nodes A, E and F are leaves. The leaf declares that it is a child, to a node connected to its
15 connected port at step S1604. In this case, the leaf recognizes that the connected port is a "parent port" which is a communication port connected to the parent node.

[0075]

20 The parent-child declaration is first performed between a leaf which is the end of a network and a "branch" with the number of connected ports of "2" or more, and then performed between branches. The parent-child relationship between communication ports is
25 sequentially determined earlier for a communication port which makes declaration earlier. A communication port which has declared that it is a child is

recognized as a "parent port", and a communication port which has received the declaration is recognized as a "child port" which is a communication port connected to a child node. For example, in Figure 14, the nodes A, E and F declare parent-child relationship after recognizing that they are leaves, and "child-parent" is determined between the nodes A and B, between the nodes E and D, and between the nodes F and D.

[0076]

10 Meanwhile, a node with the number of connected ports of "2" or more recognizes itself as a branch at step S1605. The branch accepts declaration of parent-child relationship from a node connected to its connected port at step S1606. The connected port which has accepted the declaration is recognized as a "child port" as described above. The branch, after recognizing one connected port as a "child port", checks the number of connected ports for which parent-child relationship has not been determined yet (hereinafter referred to as "undefined ports") at step S1607. If the number of undefined ports is two or more, then acceptance of parent-child declaration at step S1606 is repeated.

[0077]

25 When the number of undefined ports is "1" or less, the branch recognizes, if the number of undefined ports is "1" from determination at step S1608, the

communication port as a "parent port" at step S1609,
and declares to a node connected to the communication
port that it is a child. A branch cannot declare
parent-child relationship until the number of undefined
5 ports becomes "1". For example, in Figure 14, the
nodes B, C and D recognize that they are branches and
accept declaration of parent-child relationship from
leaves or other branches. The node D can declare
parent-child relationship to the node C after parent-
10 child relationship between D and E and between D and F
is determined. Then, the node C which has accepted
declaration of parent-child relationship from the node
D can declare parent-child relationship to the node B.
[0078]

15 If there is no undefined ports when determination
is made at step S1608, that is, if all the connected
ports provided for the branch are "child ports", then
the branch recognizes itself as a "root" at step S1610.
For example, in Figure 14, the node B, in which all the
20 connected ports are parent ports, are recognized by
other nodes as a root for making arbitration for
communication on the 1394 network. Though Figure 14
shows an example in which the node B is determined as a
root, another branch or leaf may be a root depending on
25 the timing when the node B declares parent-child
relationship. That is, there is a possibility that any
node may become a root depending on connection

configuration and timing of declaration of parent-child relationship, and the same node does not always become a root even if the connection configuration is the same.
[0079]

5 When parent-child relationship among all the connected ports has been declared as described above, each node can recognize the connection configuration of the 1394 network as a hierarchical structure (tree structure) at step S1611. A parent node is positioned
10 higher in the hierarchical structure, and a child node is positioned lower in the hierarchical structure.
[0080]

- Assignment of a node ID

 Figures 17A and 17B are flowcharts showing setting
15 of a node ID in step S1505, that is, a process for assigning a node ID to each node in detail. Figure 17A shows a process for a root, and Figure 17B shows a process for nodes other than a root. A node ID is configured by a bus number and a node number as
20 described above. In this embodiment, it is assumed that the nodes exist on the same bus and that the same bus number is assigned to each node.
[0081]

 At step S1701, a root permits setting of a node ID
25 to a node connected to a communication port with the smallest port number, among child ports to which nodes without set node ID's are connected. Next, at step

S1702, the root determines whether or not node ID's have been set for all the nodes connected to the child ports, and the step S1701 is repeated if there is any node for which a node ID has not been set. That is,
5 after node ID's are set for all the nodes connected to the communication port with the smallest port number, the root considers the child port as a port for which setting has been completed. Then, the root performs similar control for a node connected to a communication
10 port with the next smallest port number.

[0082]

When node ID's have been set for all the nodes connected to child ports finally, the root sets a node ID for itself at step S1703. Then, at step S1704, the
15 root broadcasts a self-ID packet to be described later. Basically, 0, 1, 2, ... are sequentially assigned as a node number included in a node ID to leaves and branches in that order. Accordingly, the root has the largest node number.

20 [0083]

A node which has been permitted by the root to set a node ID determines whether or not there is any child port including a node for which a node ID has not been set, at step S1711. If there is such a child port,
25 then a node connected to the child port is permitted to set a node ID at step S1712. In this case, the node permitted to set a node ID also performs the process in

Figure 17B.

[0084]

Then, at step S1713, the node again determines whether or not there is any child port including a node
5 for which a node ID has not been set. If it is determined at step S1711 or S1713 that there is not a child port including a node for which a node ID has not been set, then the node sets a node ID for itself at step S1714. Then, at step S1715, the node broadcasts a
10 self-ID packet including its node number, information about connection condition of communication ports, and the like.

[0085]

Broadcast is to transfer a communication packet of
15 a node to all of an unspecified number of other nodes constituting the 1394 network. By receiving a self-ID packet, each node can recognize the node number assigned to each node and know node numbers which can be assigned to itself.

20 [0086]

For example, in Figure 14, the node B, which is a root, permits the node A connected to a communication port with the smallest port number #0 to set a node ID. The node A assigns "0" as its node number, sets the
25 node ID, and then broadcasts a self-ID packet including the node ID.

[0087]

Next, the root permits the node C connected to a communication port with the port number #1 to set a node ID. The node C permits the node D connected to a communication port with the port number #2 to set a node ID, and node D permits the node E connected to a communication port with the port number #0 to set a node ID. When a node ID is set for the node E, the node D permits the node F connected to a communication port with the port number of #1 to set a node ID. Though subsequent description is omitted, node ID's are set for all the nodes in this procedure.

[0088]

- Self-ID packet

Figure 18 shows an example of configuration of a self-ID packet.

[0089]

Reference numeral 1801 denotes a field in which the node number of a node which has sent the self-ID packet is stored. Reference numeral 1802 denotes a field in which information about supported transfer rates is stored. Reference numeral 1803 denotes a field indicating whether a bus management function is provided or not (such as whether or not a bus manager capability is provided). Reference numeral 1804 denotes a field in which information about power consumption and supply characteristics. Reference numerals 1805 to 1807 denote fields in which

information about connection condition (connection, disconnection, parent-child relationship between communication ports, and the like) of communication ports with the numbers of #0 to #2, respectively.

5 [0090]

If a node which sends a self-ID packet has a capability to become a bus manager, the contender bit shown in the field 1803 is set to "1", and otherwise, the contender bit is set to "0".

10 [0091]

- Bus manager

A bus manager is a node which performs management described below based on various information included in the above-described self-ID packet. A node to be a
15 bus manager can perform bus management of the entire 1394 network by means of the functions.

(1) Bus power source management: Manages information for each node of the entire 1394 network about whether power can be supplied via a communication
20 cable or not, whether power supply is necessary or not, and the like.

(2) Management of rate information: Manages the maximum transfer rate among nodes based on information about transfer rates which can be supported by each
25 node.

(3) Management of topology map information:
Manages network connection configuration based on

parent-child relationship information about communication ports.

(4) Optimization of the bus based on the topology map information.

5 (5) Provision of the above information to other nodes.

[0092]

After node ID's are set, if multiple nodes have the capability of a bus manager, then a node with the
10 largest node number becomes a bus manager. Accordingly, if a root which has the largest node number has the capability of a bus manager, then the root becomes a bus manager. However, if the root does not have the capability of a bus manager, then a node which has the
15 largest node number next to that of the root and the capability of a bus manager becomes a bus manager.

[0093]

Which node has become a bus manager can be grasped by checking the contender bit 1803 of a self-ID packet
20 broadcast by each node.

[0094]

[Arbitration]

Figure 19 illustrates arbitration in the network configuration shown in Figure 2.

25 [0095]

In a 1394 network, arbitration for the bus use right is inevitably performed in prior to data transfer.

A 1394 network is a logical bus topology network, and it is possible, by relaying a packet transferred from each node to other nodes, to transfer the same packet to all the nodes in the network. Accordingly,

5 arbitration is inevitably required in order to prevent collision of packets. By arbitration, it is possible that one node can perform data transfer at a timing.

[0096]

Figure 19(a) shows that the nodes B and F request
10 the bus use right. When arbitration starts, each of the nodes B and F requests the bus use right from its parent node. The node C, the parent node which has accepted the request by the node B, relays the request for the bus use right to the node D, which is the
15 parent node of the node C and the root. That is, the request for the bus use right is eventually delivered to the root which performs arbitration.

[0097]

The root which has received the request for the
20 bus use right determines which node the bus use right should be given. It is only the root that can perform arbitration, and a node which wins the arbitration is given the bus use right.

[0098]

25 Figure 19(b) shows that the node F has been given the bus use right, and the request by the node B has been refused. The root sends a DP (Data Prefix) packet

to the node which loses the arbitration to notify that its request has been refused. The node the request of which has been refused again requests the bus use right in the next arbitration, and waits for use of the bus
5 (data transfer) until the bus use right is given.

[0099]

In this way, the root manages use of the bus of the 1394 network by performing arbitration.

[0100]

10 [Communication cycle]

It is possible for the isochronous transfer mode and the asynchronous transfer mode to co-exist in each communication cycle period by time sharing. One communication cycle period is commonly 125 μ s. Figure
15 20 illustrates that the isochronous transfer mode and the asynchronous transfer mode co-exist in one communication cycle.

[0101]

Isochronous transfer is performed prior to
20 asynchronous transfer. The reason is that idle time required for activating asynchronous transfer (subaction gap) after a cycle start packet (CSP) is set longer than idle time required for activating isochronous transfer (isochronous gap). Accordingly,
25 isochronous transfer is performed prior to asynchronous transfer.

[0102]

When each communication cycle starts, a cycle start packet (CSP) is transferred from a predetermined node. Each node can perform timekeeping the same as the other nodes by adjusting timing with the use of this CSP.

[0103]

[Isochronous transfer mode]

In the isochronous transfer mode, isochronous-type data transfer is performed. The isochronous transfer can be performed during a predetermined period after a communication cycle starts. Furthermore, in the isochronous transfer mode, isochronous transfer is inevitably performed in each cycle in order to maintain real-time transfer.

[0104]

Furthermore, the isochronous transfer mode is a transfer mode especially suitable for transfer of data requiring real-time transfer, such as motion image data or sound data including voices. The isochronous transfer mode is not for one-to-one communication but for broadcast communication, unlike the asynchronous transfer mode. That is, a packet sent from a node and isochronously transferred is equally transferred to all nodes on a network. In isochronous transfer, ack (a return code for acknowledgement) does not exist.

[0105]

In Figure 20, each of channels e, s and k

indicates a period for which each node performs isochronous transfer. In order to differentiate multiple different isochronous transfers, the 1394 interface gives a different channel number to each of
5 them. This enables isochronous transfers by multiple nodes. However, these channel numbers do not identify sending destinations but are simply logical numbers given to data.

[0106]

10 Isochronous gaps shown in Figure 20 indicate the idle state of the bus. After this idle state continues for a predetermined time, a node which desires isochronous transfer determines that it can use the bus and requests the bus use right.

15 [0107]

Figure 21 shows the format of a packet to be isochronously transferred. Hereinafter, the packet to be isochronously transferred is called an "isochronous packet". An isochronous packet is configured by a
20 header portion 2101, a header CRC 2102, a data portion 2103 and data CRC 2104.

[0108]

In the header portion 2101, there are a field 2105 in which the data length (data_length) of the data
25 portion 2103 is stored, a field 2106 in which format information (tag) about the isochronous packet is stored, a field 2107 in which the channel number

(channel) of the isochronous packet is stored, a field 2108 in which a transaction code (tcode) for identifying the format of the packet and processing to be performed is stored, and a field 2109 in which a
5 synchronization code (sy) is stored.

[0109]

[Asynchronous transfer mode]

In the asynchronous transfer mode, asynchronous-type data transfer is performed. The asynchronous
10 transfer can be performed after an isochronous transfer period ends and before the next communication cycle starts, that is, before the next CSP is transferred.

[0110]

In Figure 20, the first subaction gap indicates
15 the idle state of the bus. After the idle time reaches a predetermined value, a node which desires asynchronous transfer determines that it can use the bus and requests the bus use right. A node which acquires the bus use right as a result of arbitration
20 sends a packet to be asynchronously transferred to a predetermined node. The node which has received this packet returns ack (a return code for acknowledgement) or a response packet after an ack gap.

[0111]

25 Figure 22 shows the format of a packet to be asynchronously transferred. Hereinafter, the packet to be asynchronously transferred is called an

"asynchronous packet". The asynchronous packet is configured by a header portion 2201, a header CRC 2202, a data portion 2203 and data CRC 2204.

[0112]

5 In the header portion 2201, there are a field 2205 in which the node ID (destination_ID) of a destination node is stored, a field 2206 in which the node ID (source_ID) of a source node is stored, a field 2207 in which a label (tl) in which a series of transactions is
10 stored, a field 2208 in which a code (rt) indicating a retransmission status is stored, a field 2209 in which a transaction code (tcode) for identifying the format of the packet and processing to be performed is stored, a field 2210 in which priority order (pri) is stored, a
15 field 2211 in which a destination memory address (destination_offset) is stored, a field 2212 in which the data length (data_length) of a data portion is stored, and a field 2213 in which an extended transaction code (extended_tcode) is stored.

20 [0113]

 Asynchronous transfer is for one-to-one communication from a source node to a destination node. Though a packet sent from a source node is delivered to each node in a network, each node ignores packets
25 showing addresses other than the address of the node itself. Accordingly, only a destination node can read the packet.

[0114]

If time to transfer the next CSP arrives while asynchronous transfer is being performed, the transfer is not forcedly terminated, and the next CSP is sent
5 after the transfer ends. Accordingly, when one communication cycle continues for more than 125 μ s, the period of the next communication cycle is shortened by the time corresponding to the excess. Thereby, the 1394 network can maintain an almost constant
10 communication cycle.

[0115]

[Printer]

Figure 23 is a block diagram showing an example of internal configuration of the printer 102 shown in
15 Figure 1, which is a printer apparatus provided with an ink-jet type print head 2307.

[0116]

A CPU 2301 controls the printer 102 in accordance with an execution program stored in a ROM 2303. A RAM
20 2302 is an internal memory of the printer 102, in which there exist a receiving area for temporarily storing image data or print data to be inputted to the printer 102 via an interface, a data area for storing data for discharging ink of each of CMYK colors suitable for
25 print head 2507, which has been converted from print data, and a work area to be used by the CPU 2301 for data processing. Each block within the printer 102

performs various data transfers, controls and processings via a system bus in the printer 102.

[0117]

The basic operation of the printer 102 will be described. The CPU 2301 drives a motor 2306 via a printer controller 2304 and a printer driver 2305 and controls a carrier section on which the print head 2307 is mounted and a paper-feed mechanism. The CPU 2301 also reads the data for discharging ink from the RAM 2302 and sends the data to the printer controller 2304 and drives the print head 2307 via the printer driver 2305 to perform printing.

[0118]

The printer 102 is also provided with a LINK chip 2308 and PHY chip 2309 constituting the 1394 interface as an external interface. Accordingly, it is possible to print a visible image which is based on image data or print data inputted from an external device via the 1394 network on recording paper.

[0119]

[Digital camera]

Figure 24 is a block diagram showing an example of internal configuration of the digital camera 101 shown in Figure 1, in which the configuration related to image-taking is omitted, and only the configuration related to image data conversion processing and data transfer is shown.

[0120]

A CPU 2401 controls each block of the digital camera 101 in accordance with a program stored in a ROM 2403. The CPU 2401 performs various controls and image processings using a RAM 2402 as a work area, and performs data transfer using the RAM 2402 as a temporary storage memory.

[0121]

A camera controller 2406 performs equipment control required for image taking. It reads an image from a CCD, displays a taken image on an LCD, and performs focus and exposure setting for image taking. Data of a taken image is stored in a memory card 2408, for example, as JPEG-compressed data. Commonly, since a memory card cannot be directly connected to a system bus for access thereto, it is connected to the system bus via a card controller 2407 for controlling reading and writing of an ATA-type memory card.

[0122]

The digital camera 101 is also provided with a 1394 interface as an external interface and can send image data and print data to an external device via a LINK chip 2404 and a PHY chip 2405.

[0123]

[Data processing]

Figure 25 is a block diagram showing a process for converting image data which has been taken by the

digital camera 101 and JPEG-compressed to print data
suitable for the printer 102.

[0124]

Since it is necessary to expand compressed data in
5 order to perform image processing, JPEG data is
expanded by a JPEG expansion section 251 and converted
to RGB data.

[0125]

Correction processing about image contrast,
10 brightness, gamma, saturation, color cast, outline and
the like are performed for the RGB data by an image
correction section 252, and the RGB data becomes RGB'
data which is corrected data.

[0126]

15 The RGB' data is converted by a color processing
section 253 to CMY data indicating cyan, magenta and
yellow suitable for the color space determined based on
the ink used by the printer 102.

[0127]

20 A black component is extracted by a UCR section
254, and the CMY data is converted to CMYK data with
four CMYK colors in which data indicating black has
been added to CMY. "UCR" is an abbreviation of "Under
Color Removal".

25 [0128]

Furthermore, it is necessary to binarize or
quantize the multi-value CMYK data to data indicating

discharge of ink in accordance with the inkjet print method, which is the print method of the printer 102. A half-toning section 255 converts the multi-value CMYK data to CMYK data suitable for the resolution of the printer 102. That is, by utilizing pseudo-gradation processing such as error diffusion and dithering, the multi-value CMYK data is converted to CMYK data with a resolution suitable for the printer 102 and with two, three or four values. Figure 25 shows an example of conversion to CMYK binary data.

[0129]

This CMYK binary data is eventually converted by the printer 102 to discharge pattern data suitable for the structure of the print head 2307, and the print head 2307 is driven.

[0130]

To determine performance of image processing, it is necessary to measure data processing time. Table 1 shows an example of processing time required for performing the image processing shown in Figure 25 for sample JPEG data, which is a predetermined image, by means of the digital camera 101 and the printer 102. Though the sample image to be used is not especially limited, it is conceivable that measurement error is rarely occurred in using JPEG data of a colored image rather than using JPEG data of a monochrome image.

[0131]

Table 1 shows processing time measured by causing the digital camera 101 and the printer 102 to process sample JPEG data with the size of 100 KB. In Table 1, sample image data processing time is shown for each processing section. If the time required for the processing is immediately used as a performance value, it indicates that the processing capacity is higher as the numerical value is smaller. This can be also used as an indication of processing time.

10 [0132]

[Table 1]

Processing section	Data size (KB)	Processing time (ms)	
		camera	printer
JPEG data	100	-	-
JPEG expansion section	300	60	120
Image correction section	300	60	60
Color processing section	450	60	40
UCR section	450	30	33
Half-toning section	800	2000	1000

If at least the processing capacity of the device itself (as shown in Table 1) is stored in a ROM in

advance, then, by providing the processing capacity of the digital camera to the printer or providing the processing capacity of the printer to the digital camera via the 1394 network, it is possible for the CPU of the digital camera or the printer to determine how a series of image processings related to printing should be shared by the digital camera and the printer from the information.

[0133]

10 Table 2 shows the result of estimation of processing time required for processing and printing JPEG data with the size of 10 MB, which has been performed with the use of the performance values shown in Table 1.

15 [0134]

[Table 2]

Processing section	Data size (MB)	Processing time (ms)	
		camera	printer
JPEG data	10	-	-
JPEG expansion section	30	6	12
Image correction section	30	6	6
Color processing section	45	6	4
UCR section	45	3	3.3

Half-toning section	80	200	100
------------------------	----	-----	-----

If it is assumed that the processing time of each processing section is approximately in simple proportion to data size, the processing time in Table 2 is to be one hundred times the processing time in Table 1 because the size of the sample JPEG data in Table 1 is 100 KB and the size of the sample JPEG data in Table 2 is 10 MB, which is one hundred times 100 KB. The data size shown in Table 2 is also calculated on the assumption that it is in simple proportion.

[0135]

Next, description will be made on a method for selecting a processing section with the use of the performance values shown in Table 1, in other words, a method for causing the digital camera 101 and the printer 102 to share processing, with reference to Figure 26.

[0136]

A device to determine sharing of processing acquires data about processing capacity as shown in Table 1 from a device connected via a 1394 network 103. That is, if the CPU 2401 of the digital camera 101 is to determine sharing of processing, then the CPU 2401 acquires data about processing capacity from the printer 102, and if the CPU 2301 of the printer 102 is

to determine sharing of processing, then the CPU 2301 acquires data about processing capacity from the digital camera 101. Which device should determine sharing of processing is first based on whether a
5 program for determining sharing of processing is stored in the ROM. Second, if such a program is stored in the ROM of both devices, then, for example, a device which is a root may perform the determination.

[0137]

10 If information about processing capacity once acquired is stored in a non-volatile memory, it is not necessary to re-acquire the data about processing capacity every time sharing of processing is determined as far as the connected device is not replaced.

15 [0138]

Next, at step S2, the device to determine sharing of processing determines sharing of processing in accordance with a procedure described below.

[0139]

20 In determining sharing, when processing sections with a shorter processing time are selected first, the JPEG expansion section 251, the image correction section 252 and the UCR section 254 of the digital camera 101 are selected, and the image correction
25 section 252, the color processing section 253 and the half-toning section 255 of the printer 102 are selected.

[0140]

As for processing requiring almost the same processing time from both devices, like the processing by the image correction section 252, it is possible to eliminate necessity of data transfer, processing prior to the processing, by performing the processing on the side of a device which has performed the immediately previous processing. Therefore, as for the image correction section 252, it is desirable to select the image correction section 252 of the digital camera 101. Similarly, if processing is performed by the two devices alternately, time required for data transfer, processing prior to the actual processing, must be considered. That is, the device to perform processing should not be switched as far as possible in order to minimize delay in processing time due to the time required for data transfer.

[0141]

The speed of approximately 40 MB/s can be obtained as the data transfer rate of a 1394 network in the case of the S400 standard. However, if it is assumed that the actual speed is reduced to the half thereof when overhead required for protocol negotiation is added, the speed will be about 20 MB/s. Accordingly, it takes two seconds or more to send CMY data with the size of 45 MB (see Table 2) for which processing by the color processing section 253 of the printer 102 has been performed to the UCR section 254 of the digital camera

101 which performs high-speed processing. Since
difference between processing times of the UCR sections
254 of both devices is 0.3 seconds, it is more
efficient to continue using the UCR section 254 of the
5 printer 102 to perform processing if the color
processing section 253 of the printer 102 is used.
[0142]

If the processing sections are selected so that
the processing time is the shortest, based on the
10 result of the above examination, the selection is as
follows. The JPEG expansion section 251 of the digital
camera 101 --> the image correction section 252 --
(data transfer) --> the color processing section 253 of
the printer 102 --> the UCR section 254 --> the half-
15 toning section 255. The total processing time required
for processing JPEG data with the size of 10 MB is
estimated to be $6+6+1.5+4+3.3+100=120.8$ seconds.
[0143]

By comparison, in the case where all the
20 processings are performed by the digital camera 101,
data transfer is performed after the processing by the
half-toning section 255 is performed, and the total
processing time required for processing JPEG data with
the size of 10 MB is estimated to be $6+6+6+3+200+4=225$
25 seconds. A delay of $225-120.8=104.2$ seconds is
expected to occur. In the case where all the
processings are performed by the printer 102, the

processings are performed after data transfer, and the total processing time required for processing JPEG data with the size of 10 MB is estimated to be $0.5+12+6+4+3.3+100=125.8$ seconds. A delay of $125.8-120.8=5$ seconds is expected to occur.

[0144]

That is, if processing sections are optimally selected, image processings for JPEG data with the size of 10 MB is estimated to be finished about 104 seconds earlier than the case of causing the digital camera 101 to perform all the image processings and about five seconds earlier than the case of causing the printer 102 to perform all the image processings. If the CPU of the digital camera 101 or the printer 102 determines sharing of a series of image processings based on such a determination result, and another device complies with the determination, it is possible to appropriately set sharing of image processings enabling the processing time to be the shortest when direct print is to be performed in combination with a digital camera and a printer.

[0145]

The device which has determined sharing of processings as described above notifies the counterpart device of the sharing of processings via the 1394 network 103, at step S3. The device which has received this notification determines processings to be

performed by itself and the form of data to be sent or received in accordance with the notified sharing of processings.

[0146]

- 5 If sharing of processings once determined is stored in a non-volatile memory, it is not necessary to re-determine sharing of processings as far as the connected device is not replaced.

[0147]

- 10 In this way, according to the embodiment, when direct print is performed between an image supply device such as the digital camera 101 and a printing device such as the printer 102 connected via the 1394 serial bus 103, it is possible to appropriately share
15 image processings between the devices according to the processing capacity of each image processing function of each device and the relation between the data size and the data transfer rate, and thereby efficient direct print can be performed.

20 [0148]

[Other embodiments]

- The present invention may be applied to a system configured by multiple pieces of equipment (for example, a host computer, interface equipment, a reader, a
25 printer and the like) or to an apparatus configured by one piece of equipment (for example, a copying machine, a facsimile machine and the like).

[0149]

Needless to say, the object of the present invention can be achieved by providing a storage medium (or a recording medium) in which program codes of software for realizing the above-described functions of the embodiment are stored for a system or an apparatus, and a computer (or a CPU or a MPU) of the system or the apparatus reading and executing the program codes stored in the recording medium. In this case, the program codes themselves, which have been read from the storage medium, realize the above-described functions of the embodiment, and the storage medium in which the program codes are stored constitutes the present invention. The above-described functions of the embodiment are not realized only by executing the program codes read by the computer. Needless to say, the case is also included where an operating system (OS) running on the computer performs a part or all of the actual processings based on the instructions of the program codes, and the above-described functions of the embodiment are realized by the processings.

[0150]

Furthermore, needless to say, the case is also included where the program codes read from the storage medium is written in the memory provided for a feature expansion card inserted in the computer or a feature expansion unit connected to the computer, then a CPU or

the like provided for the feature expansion card or the feature expansion unit performs a part or all of the actual processings, and then the above-described functions of the embodiment are realized by the processings based on the instructions of the program codes.

[0151]

When the present invention is applied to the above-described storage medium, program codes corresponding to the flowcharts described before are stored in the storage medium.

[0152]

[Effects of the invention]

As described above, according to the present invention, efficient image processings can be performed by causing individual devices to share the image processings.

[Brief Description of the Drawings]

[Figure 1]

A diagram showing an example of common configuration of a system to which the present invention is applied.

[Figure 2]

A diagram showing an example of configuration of a 1394 network constituted by nodes provided with a 1394 interface.

[Figure 3]

A diagram illustrating components of the 1394 interface.

[Figure 4]

A diagram showing services which can be provided
5 by a link layer.

[Figure 5]

A diagram showing services which can be provided
by a transaction layer.

[Figure 6]

10 A diagram illustrating address space in the 1394 interface.

[Figure 7]

A diagram showing addresses and functions of
information stored in a CSR core register.

15 [Figure 8]

A diagram showing addresses and functions of
information stored in a serial bus register.

[Figure 9]

A diagram showing configuration of a minimum-form
20 configuration ROM.

[Figure 10]

A diagram showing configuration of a general-form
configuration ROM.

[Figure 11]

25 A diagram showing addresses and functions of
information stored in a serial bus apparatus register
in a unit space.

[Figure 12]

A sectional view of a communication cable in conformity with the IEEE 1394 standard.

[Figure 13]

5 A diagram illustrating a DS-Link coding method.

[Figure 14]

A diagram illustrating a basic sequence from start of bus reset to assignment of a node ID.

[Figure 15]

10 A diagram illustrating a basic sequence from start of bus reset to assignment of a node ID.

[Figure 16]

A diagram illustrating a basic sequence from start of bus reset to assignment of a node ID.

15 [Figure 17A]

A flowchart showing a process for assigning a node ID to each node in detail.

[Figure 17B]

A flowchart showing a process for assigning a node
20 ID to each node in detail.

[Figure 18]

A diagram showing an example of configuration of a self-ID packet.

[Figure 19]

25 A diagram illustrating arbitration in the network configuration shown in Figure 2.

[Figure 20]

A diagram illustrating condition that an isochronous transfer mode and an asynchronous transfer mode co-exist in one communication cycle.

[Figure 21]

- 5 A diagram showing a format of a packet to be isochronously transferred.

[Figure 22]

A diagram showing a format of a packet to be asynchronously transferred.

- 10 [Figure 23]

A block diagram showing an example of internal configuration of a printer shown in Figure 1.

[Figure 24]

- 15 A block diagram showing an example of internal configuration of a digital camera shown in Figure 1.

[Figure 25]

A block diagram showing a process for converting image data which has been taken by a digital camera and JPEG-compressed to print data suitable for a printer.

- 20 [Figure 26]

A flowchart illustrating a method for causing a digital camera and a printer to share image processings.

Figure 2

- #1 NODE
- #2 1394 SERIAL BUS

5 Figure 3

- 301 1394 SERIAL BUS CABLE
- 302 1394 COMMUNICATION PORT
- 303 PHYSICAL LAYER
- 304 LINK LAYER
- 10 305 TRANSACTION LAYER
- 306 SERIAL BUS MANAGEMENT
- 307 APPLICATION LAYER
- #1 SOFTWARE
- #2 FIRMWARE
- 15 #3 HARDWARE

Figure 4

- #1 LINK LAYER SERVICE
- #2 REQUESTING NODE
- 20 LINK LAYER
- #3 RESPONDING NODE
- LINK LAYER
- #4 ARBITRATION, DATA PACKET
- #5 LINK REQUEST
- 25 #6 LINK NOTIFICATION
- #7 LINK RESPONSE
- #8 ACKNOWLEDGEMENT

#9 DOES NOT EXIST IN THE CASE OF BROADCAST OR
ISOCHRONOUS PACKET

#10 LINK CONFIRMATION

5 Figure 5

#1 TRANSACTION LAYER SERVICE

#2 REQUESTING NODE

TRANSACTION LAYER

#3 RESPONDING NODE

10 TRANSACTION LAYER

#4 TRANSACTION CONTROL INFORMATION (INCLUDING DATA IN
THE CASE OF WRITE/LOCK)

#5 TRANSACTION REQUEST

#6 TRANSACTION NOTIFICATION

15 #7 TRANSACTION RESPONSE

#8 TRANSACTION STATE INFORMATION (INCLUDING DATA IN
THE CASE OF WRITE/LOCK)

#9 TRANSACTION CONFIRMATION

20 Figure 6

#1 MEMORY SPACE

#2 PRIVATE

#3 REGISTER

#4 CSR CORE

25 #5 SERIAL BUS

#6 CONFIGURATION ROM

#7 UNIT SPACE

#8 BUS NUMBER
 #9 NODE NUMBER
 #10 REGISTER SPACE
 #11 REGISTER ADDRESS (0 TO FFFFFFFF)
 5 #12 PRIVATE SPACE
 #13 PRIVATE ADDRESS (0 TO FFFFFFFF)
 #14 MEMORY SPACE (0 TO FFFF DFFF FFFF)
 #15 10 BITS
 #16 6 BITS
 10 #17 20 BITS
 #18 28 BITS

Figure 7

#1 CRS CORE REGISTER
 15 #2 OFFSET (IN HEXADECIMAL)
 #3 REGISTER NAME
 #4 FUNCTION
 #5 INFORMATION ABOUT STATE AND CONTROL
 #6 INFORMATION INDICATING WHETHER STATE_CLEAR CAN BE
 20 WRITTEN OR NOT
 #7 BUS ID + NODE ID
 #8 BUS IS RESET BY WRITING IN THIS AREA
 #9 REGISTER FOR ACCESSING ROM OF MORE THAN 1K
 #10 VALUE OF TIMER FOR DETECTING TIMEOUT OF SPLIT
 25 TRANSACTION
 #11 REGISTER FOR DIAGNOSIS
 #12 NOT IMPLEMENTED FOR IEEE 1394

- #13 INTERRUPTION NOTIFICATION REGISTER
- #14 NOT IMPLEMENTED FOR IEEE 1394
- #15 MESSAGE NOTIFICATION REGISTER
- #16 RESERVATION
- 5 #17 RESERVATION FOR IEEE 1394

Figure 8

- #1 SERIAL BUS REGISTER
- #2 OFFSET (IN HEXADECIMAL)
- 10 #3 REGISTER NAME
- #4 FUNCTION
- #5 COUNTER FOR ISOCHRONOUS TRANSFER
- #6 REGISTER FOR SYNCHRONIZATION OF TIME
- #7 REGISTER RELATED TO POWER SUPPLY
- 15 #8 CONTROLS RETRY OF TRANSACTION LAYER
- #9 RESERVATION
- #10 NODE ID OF BUS MANAGER
- #11 MANAGES BAND FOR ISOCHRONOUS TRANSFER
- #12 MANAGES CHANNEL NUMBER FOR ISOCHRONOUS TRANSFER
- 20 #13 REGISTER FOR DIAGNOSIS
- #14 RESERVATION

Figure 9

- #1 VENDOR ID
- 25 #2 MINIMUM-FORM CONFIGURATION ROM

Figure 11

- #1 SERIAL BUS APPARATUS REGISTER
- #2 OFF SET (IN HEXADECIMAL)
- #3 REGISTER NAME
- #4 FUNCTION
- 5 #5 RESERVATION
- #6 SERIAL BUS CONFIGURATION INFORMATION
- #7 RESERVATION
- #8 SERIAL BUS TRANSFER RATE INFORMATION
- #9 RESERVATION

10

Figure 12

- #1 POWER SOURCE LINE
- #2 TWISTED-PAIR SIGNAL LINE
- #3 SIGNAL LINE SHIELD
- 15 #4 (SECTIONAL VIEW OF CABLE)

Figure 13

- #1 DATA
- #2 STROBE
- 20 #3 CLOCK
- #4 (CLOCK IS OBTAINED BY PERFORMING EXCLUSIVE-OR OF DATA AND STROBE)

Figure 14

- 25 #1 NODE
- #2 LEAF
- #3 BRANCH

- #4 ROOT
- #5 PORT
- #6 PARENT PORT CONNECTED TO PARENT NODE
- #7 CHILD PORT CONNECTED TO CHILD NODE

5

Figure 15

- S1501 HAS BUS RESET OCCURRED?
- S1502 DECLARES PARENT-CHILD RELATIONSHIP
- S1503 HAS PARENT-CHILD RELATIONSHIP BEEN DETERMINED
- 10 FOR ALL NODES?
- S1504 DETERMINES ROOT
- S1505 SETS NODE ID
- S1506 HAS SETTING OF NODE ID ENDED?
- S1507 DATA TRANSFER

15

Figure 16

- S1601 CONFIRMS PORT CONNECTION CONDITION AND COUNTS
- THE NUMBER OF CONNECTED PORTS
- S1602 THE NUMBER OF CONNECTED PORTS?
- 20 S1603 RECOGNIZES NODE AS LEAF
- S1604 MAKES DECLARATION OF CHILD
- S1605 RECOGNIZES NODE AS BRANCH
- S1606 ACCEPTS DECLARATION OF PARENT-CHILD RELATIONSHIP
- S1607 THE NUMBER OF UNDEFINED PORTS?
- 25 S1608 THE NUMBER OF UNDEFINED PORTS?
- S1609 MAKES DECLARATION OF CHILD
- S1610 RECOGNIZES NODE AS ROOT

(B) PERMISSION/REFUSAL OF USE OF BUS

- #1 ROOT
- #2 REQUEST
- #3 REFUSAL
- 5 #4 PERMISSION

Figure 20

- #1 CYCLE TIME 125 μ s
- #2 CHANNEL
- 10 #3 PACKET
- #4 (LONG GAP)
- #5 (SHORT GAP)

Figure 21

- 15 #1 ISOCHRONOUS PACKET

Figure 22

- #1 ASYNCHRONOUS PACKET

20 Figure 23

- 2304 PRINTER CONTROLLER
- 2304 PRINTER DRIVER
- 2306 MOTOR
- 2307 PRINT HEAD

25

Figure 24

- 2406 CAMERA CONTROLLER

2407 MEMORY CONTROLLER

2408 MEMORY CARD

Figure 25

- 5 251 JPEG EXPANSION SECTION
- 252 IMAGE CORRECTION SECTION
- 253 COLOR PROCESSING SECTION
- 254 UCR SECTION
- 255 HALF-TONING SECTION
- 10 #1 JPEG DATA
- #2 RGB DATA
- #3 RGB' DATA
- #4 CMY DATA
- #5 CMYK DATA
- 15 #6 CMYK BINARY DATA

Figure 26

- S1 ACQUIRES DATA ABOUT PROCESSING CAPACITY
- S2 DETERMINES SHARING OF PROCESSING
- 20 S3 NOTIFIES SHARING OF PROCESSING

に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることはいうまでもない。

【0151】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードが格納されることになる。

【0152】

【発明の効果】以上説明したように、本発明によれば、個々のデバイスに画像処理を分担することで効率的な画像処理を行うことができる。

【図面の簡単な説明】

【図1】本発明を適用するシステムの一般的な構成例を示す図、

【図2】1394インタフェースを備えるノードにより構成される1394ネットワークの構成例を示す図、

【図3】1394インタフェースの構成要素を説明する図、

【図4】リンクレイヤが提供可能なサービスを示す図、

【図5】トランザクションレイヤが提供可能なサービスを示す図、

【図6】1394インタフェースにおけるアドレス空間を説明する図、

【図7】CSRコアレジスタに格納される情報のアドレスおよび機能を示す図、

【図8】シリアルバスレジスタに格納される情報のアドレスおよび機能を示す図、

【図9】最小形式のコンフィグレーションROMの構成を示す図、

【図10】一般形式のコンフィグレーションROMの構成を示す図、

【図11】ユニット空間のシリアルバス装置レジスタに格納される情報のアドレスおよび機能を示す図、

*

*【図12】IEEE1394規格に準拠した通信ケーブルの断面図、

【図13】DS-Link符号化方式を説明する図、

【図14】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図15】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図16】バスリセットの開始からノードIDを割り当てるまでの基本的なシーケンスを説明する図、

【図17A】各ノードにノードIDを割り当てる処理を詳細に示すフローチャート、

【図17B】各ノードにノードIDを割り当てる処理を詳細に示すフローチャート、

【図18】セルフIDパケットの構成例を示す図、

【図19】図2に示したネットワーク構成におけるアービトレーションを説明する図、

【図20】一通信サイクルにアイソクロナス転送モードおよびアシンクロナス転送モードを混在させた状態を説明する図、

20 【図21】アイソクロナス転送されるパケットのフォーマットを示す図、

【図22】アシンクロナス転送されるパケットのフォーマットを示す図、

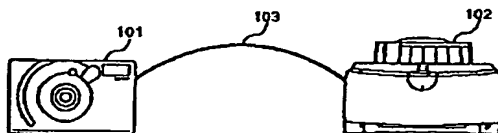
【図23】図1に示すプリンタの内部構成例を示すブロック図、

【図24】図1に示すデジタルカメラの内部構成例を示すブロック図、

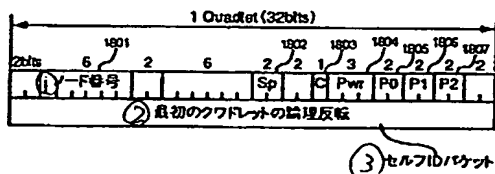
30 【図25】デジタルカメラで撮影されJPEG圧縮された画像データを、プリンタに合わせた印刷データに変換する処理を示すブロック図、

【図26】画像処理をデジタルカメラおよびプリンタに分担させる方法を説明するフローチャートである。

【図1】 Fig. 1



【図18】



【図2】 Fig. 2

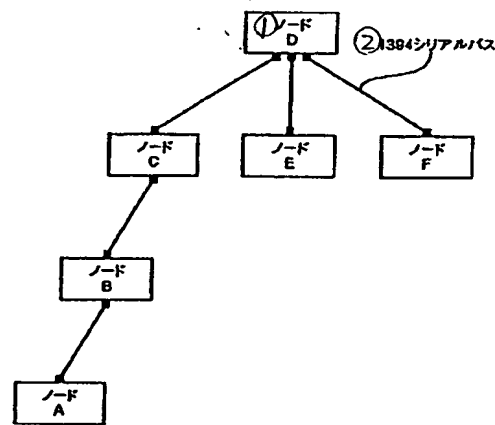
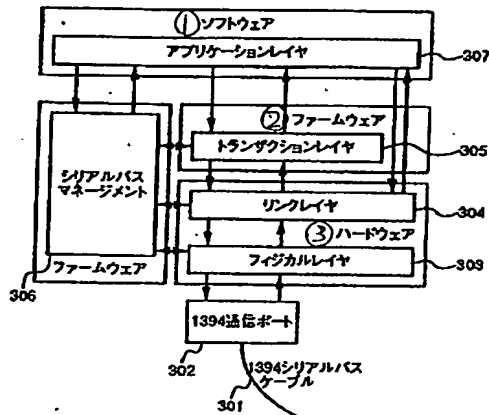
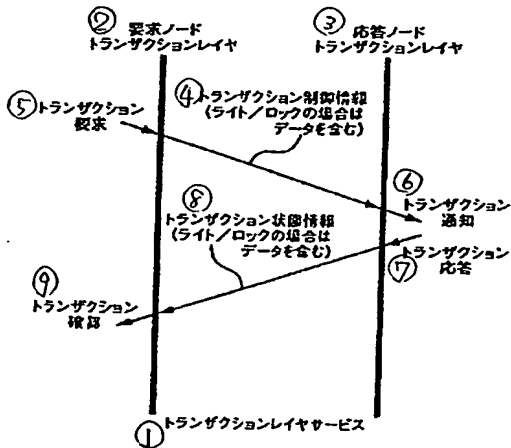
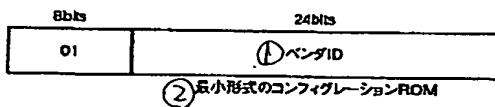
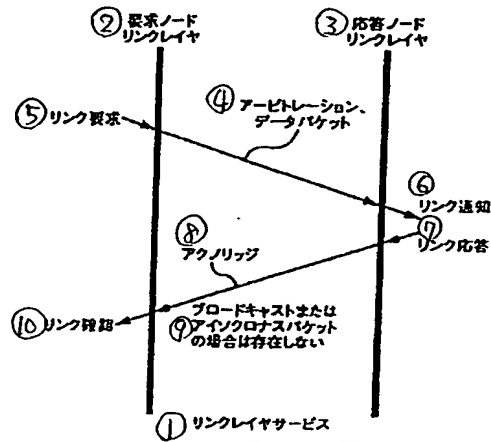


Fig. 3
【図3】Fig. 5
【図5】Fig. 9
【図9】Fig. 4
【図4】Fig. 7
【図7】

① CRSコアレジスタ			
② オフセット (16進数)	③ レジスタ名称	④ 機能	
000	STATE_CLEAR	⑤	状態と制御の情報
004	STATE_SET	⑥	STATE_CLEARの書き込み可否を示す情報
008	NODE_IDS	⑦	バスID+ノードID
00C	RESET_START	⑧	この領域に対する書き込みでバスをリセットする
010~014	INDIRECT_ADDRESS, INDIRECT_DATA	⑨	1Kより大きいROMアクセスするためのレジスタ
018~01C	SPUT_TIMEOUT	⑩	スプリットトランザクションのタイムアウトを発生するタイムの値
020~02C	ARGUMENT_TEST_START, TEST_STATUS	⑪	専用のレジスタ
030~04C	UNITS_BASE, UNITS_BOUND, MEMORY_BASE, MEMORY_BOUND	⑫	IEEE1394では実装しない
050~054	INTERRUPT_TARGET, INTERRUPT_MASK	⑬	割り込み通知レジスタ
058~07C	CLOCK_VALUE, CLOCK_TICK_PERIOD, CLOCK_STROBE_ARRIVED, CLOCK_INFO	⑭	IEEE1394では実装しない
080~0FC	MESSAGE_REQUEST, MESSAGE_RESPONSE	⑮	メッセージ通知レジスタ
100~17C		⑯	予約
180~1FC	ERROR_LOG_BUFFER	⑰	IEEE1394用に予約

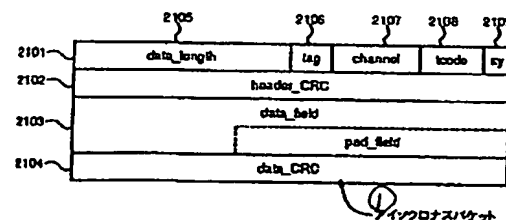
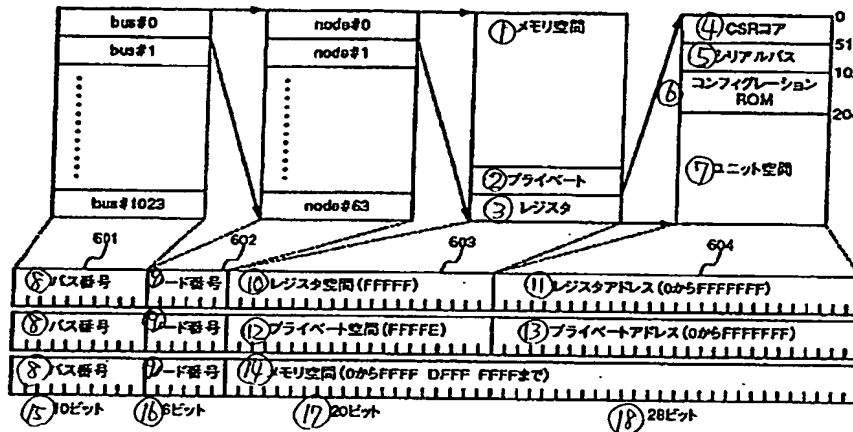
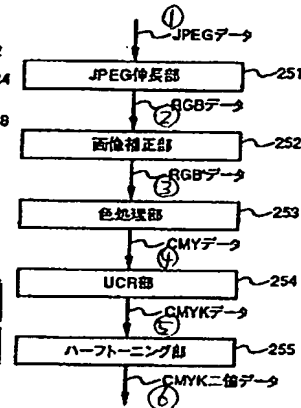
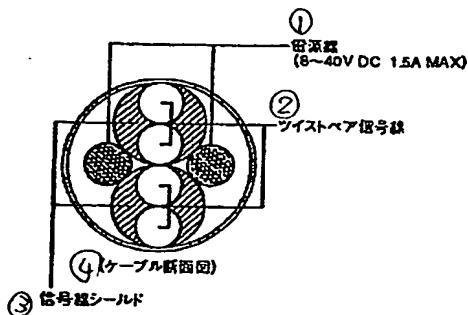
Fig. 21
【図21】

Fig. 6
【図6】Fig. 25
【図25】Fig. 8
【図8】

① シリアルバスレジスタ			
② オフセット (16進数)	③ レジスタ名称	④ 機能	
200	CYCLE_TIME	⑤ アシクロナス転送のためのカウンタ	
204	BUS_TIME	⑥ 時間を同期するためのレジスタ	
208	POWER_FAIL_IMMINENT	⑦ 電源供給に関するレジスタ	
20C	POWER_SOURCE		
210	BUSY_TIMEOUT	⑧ トランザクションレイヤの再実行を制御する	
214		⑨ 予約	
218			
21C	BUS_MANAGER_ID	⑩ バスマネージャのノードID	
220	BANDWIDTH_AVAILABLE	⑪ アシクロナス転送の帯域を管理する	
224		⑫ アシクロナス転送のチャネル番号を管理する	
228	CHANNELS_AVAILABLE		
22C	MAINT_CONTROL	⑬ 診断用レジスタ	
230	MAINT_UTILITY		
234		⑭ 予約	
3FC			

Fig. 10
【図10】

Bus Info Block Length	ROM Length	CRC
Bus Info Block		
Root Directory		
Node Dependent Info Directory		
Unit Directories		
Root & Unit Leaves		
Vendor Dependent Information		

Fig. 11
【図11】Fig. 12
【図12】

① シリアルバス装置レジスタ			
② オフセット (16進数)	③ レジスタ名称	④ 機能	
800		⑤ 予約	
FFC			
1000	TOPOLOGY_MAP	⑥ シリアルバスの構成情報	
13FC		⑦ 予約	
1400		⑧ 予約	
1FFC			
2000	SPEED_MAP	⑨ シリアルバスの転送速度の情報	
2FFC		⑩ 予約	
3000		⑪ 予約	
FFFC			

Fig. 13

【図13】

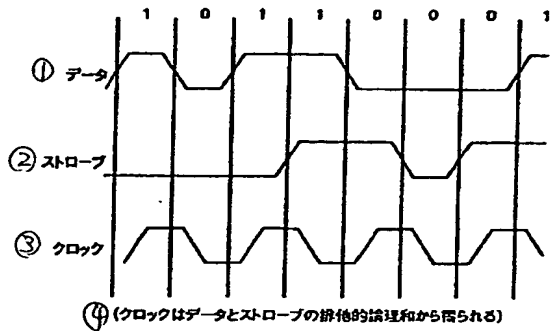


Fig. 14

【図14】

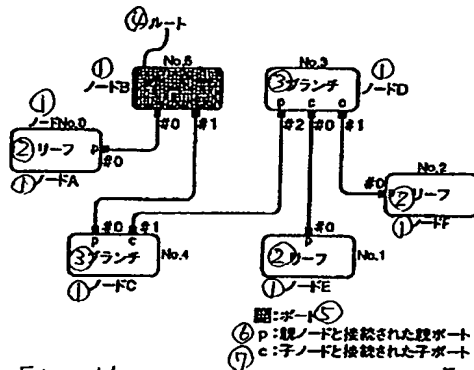


Fig. 15

【図15】

Fig. 16

【図16】

Fig. 17A

【図17A】

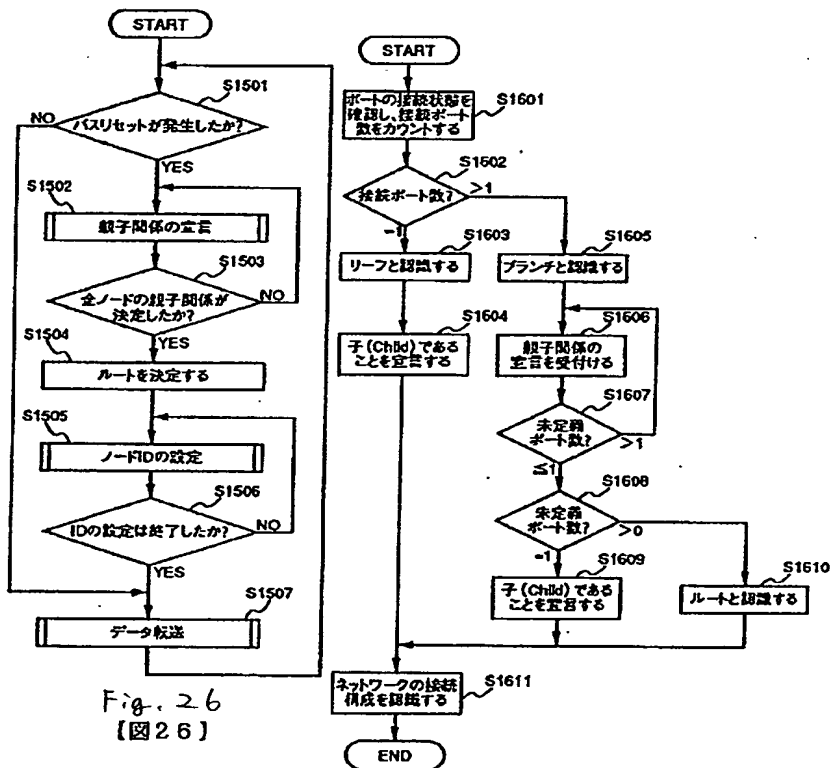


Fig. 26

【図26】

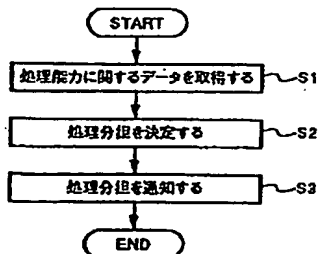
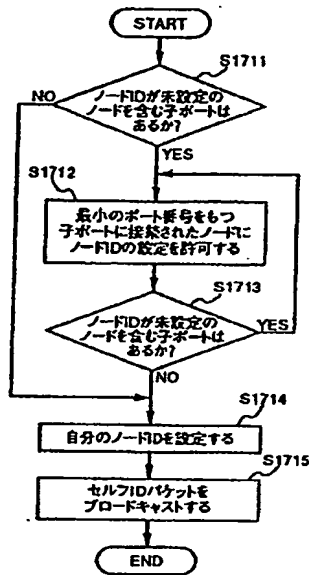
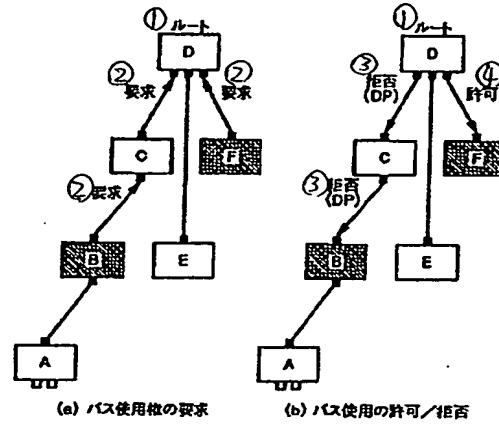
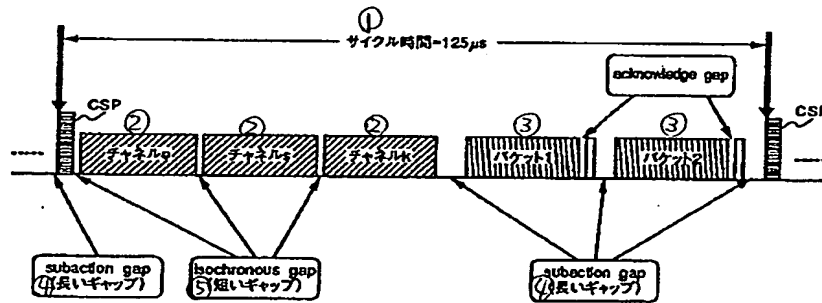
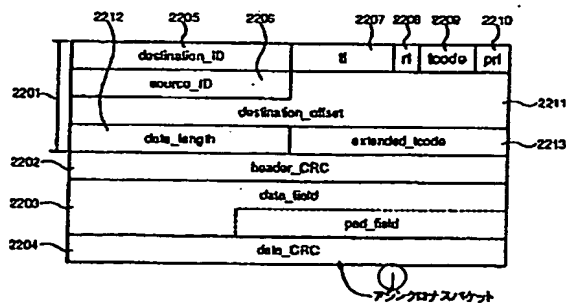


Fig. 17B
【図17B】Fig. 19
【図19】Fig. 20
【図20】Fig. 22
【図22】

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.